



TpaCutOEM

TpaCutOEM Library

Manuale d'uso

v. 1.1.0

05/02/2024





1. PRESENTAZIONE	5
1.1. VERSIONI.....	5
1.2. LICENZA	5
1.3. PROPRIETÀ E DIRITTI D’AUTORE	5
2. GUIDA UTENTE	5
2.1. OTTIMIZZAZIONE DI TAGLIO	6
2.2. UNITÀ E COORDINATE	7
2.3. DIREZIONE E VERTICE DI SVILUPPO	7
2.4. GRUPPI E FILTRI DI CORRISPONDENZA	7
2.4.1. <i>Il controllo della Venatura</i>	8
2.5. I TAGLI	8
2.5.1. <i>Livelli di tagli</i>	9
2.5.2. <i>Rifili</i>	11
2.5.3. <i>Taglio di rilascio della tensione</i>	11
2.5.4. <i>Margini del foglio</i>	11
2.6. L’OTTIMIZZAZIONE IN TPA_C.....	12
2.6.1. <i>Ottimizzazioni progressive</i>	12
2.6.2. <i>Criteri e priorità di ottimizzazione</i>	13
2.6.2.1. Utilizzo delle lastre	13
2.6.2.2. Utilizzo dei rettangoli.....	13
I piazzamenti extra.....	14
2.7. SUPPORTO PER LA GESTIONE DEI PROGETTI DI TAGLIO	14
2.8. LIMITI CONOSCIUTI DELLA LIBRERIA	14
3. GUIDA ALLE FUNZIONI DELLA LIBRERIA	14
3.1. GESTIONE DELLA LICENZA.....	14
3.1.1. <i>IsValidLicense</i>	14
3.2. COSTANTI	14
3.3. ENUMERAZIONI	15
3.3.1. <i>CutError</i>	15
3.4. STRUTTURE	15
3.4.1. <i>OneRect</i>	15
3.4.2. <i>OneSheet</i>	16
3.5. DEFINIZIONE DELLE FUNZIONALITÀ.....	17
3.5.1. <i>Funzionalità generali</i>	17
3.5.1.1. Version.....	17
3.5.1.2. LastError	17
3.5.1.3. ErrorMessage.....	18
3.5.2. <i>Funzionalità corrispondenti a settaggi generali di ottimizzazione di taglio</i>	18
3.5.2.1. NumberCustom	18
3.5.2.2. NumberRectParam	18
3.5.2.3. EnableRectEdge	18
3.5.2.4. OrderSheetDim	18
3.5.2.5. OrderBeforeScrap	18
3.5.2.6. OrderStrips	18
3.5.2.7. OrderSubStrips	19
3.5.3. <i>Funzionalità relative alle assegnazioni generali di un progetto</i>	19
3.5.3.1. Unit	20

3.5.3.2.	CutterDiameter	20
3.5.3.3.	Direction	20
3.5.3.4.	Corner	20
3.5.3.5.	MaxCutLevels	20
3.5.3.6.	TensionGap	20
3.5.3.7.	PreCut	20
3.5.3.8.	Longcut	20
3.5.3.9.	Transvcut	21
3.5.3.10.	Zcut	21
3.5.3.11.	Custom1, Custom2, ..., Custom10	21
3.5.3.12.	ClearAll	21
3.6.	DEFINIZIONE DEI RETTANGOLI	21
3.6.1.	AddRct	21
3.6.2.	ClearRct	21
3.6.3.	CountRect	21
3.6.4.	ReadRect	22
3.6.5.	ReadRectIndex	22
3.7.	DEFINIZIONE DEI FOGLI	22
3.7.1.	AddSheet	22
3.7.2.	ClearSheet	23
3.7.3.	CountSheet	23
3.7.4.	ReadSheet	23
3.7.5.	ReadSheetIndex	23
3.8.	SOLUZIONE DI TAGLIO	23
3.8.1.	Compute	23
3.9.	I RISULTATI DELL'OTTIMIZZAZIONE	24
3.9.1.	NumberOfSolutions	24
3.9.2.	SelectSolution	24
3.9.3.	SolutionSheets	24
3.9.4.	GetSheetId	24
3.9.5.	NumberOfRepetitions	25
3.9.6.	UsedSheets	25
3.9.7.	UsedRects	25
3.9.8.	FitnessSheet	26
3.9.9.	ReadResolRect	26
3.9.10.	NumberOfCuts	26
3.9.11.	ReadResolCut	27
3.9.12.	CutLinear	28
3.9.13.	CutArea	28
3.9.14.	MarginArea	28
3.9.15.	GetWaste	29
3.9.16.	EstimatedTime	29
3.9.17.	EstimatedCost	29
3.9.18.	Export	30
	Struttura di file corrispondente ad uno schema di taglio	30
	Nodo <GENERALSETTINGS>	31
	Nodo <TECHSETTINGS>	31
	Nodo <DIM>	31
	Nodo <DIMTRIMS>	31
	Nodo <DATA>	31

Nodo <PIECESLIST>	31
Nodo <DRAW>	32
3.10. FUNZIONI DI SERIALIZZAZIONE DEL PROGETTO	36
3.10.1. <i>SaveProject</i>	36
3.10.2. <i>LoadProject</i>	36
3.10.3. <i>ImportProject</i>	36
4. GUIDA ALL'UTILIZZO DELLA LIBRERIA.....	38
4.1. DIAGRAMMA DI FLUSSO TIPICO	38
4.2. VERIFICHE PRELIMINARI	38
4.3. ASSEGNAZIONE DEI SETTAGGI DI FUNZIONAMENTO GENERALE	39
4.4. ASSEGNAZIONE DEI SETTAGGI GENERALI DI PROGETTO	39
4.5. ASSEGNAZIONE DEI RETTANGOLI.....	39
4.6. ASSEGNAZIONE DEI FOGLI	39
4.7. ESECUZIONE DELL'OTTIMIZZAZIONE DI TAGLIO	39
4.7.1. <i>Acquisire il risultato della soluzione</i>	40
4.7.1.1. Codice di esempio: come acquisire le informazioni generali dei fogli della soluzione	40
4.7.1.2. Codice di esempio: ciclo di acquisizione dei piazzamenti di un foglio	40
4.7.1.3. Codice di esempio: ciclo di acquisizione dei tagli di un foglio	41
4.7.2. <i>Gestire soluzioni multiple</i>	41
4.7.2.1. Codice di esempio: navigazione tra soluzioni multiple	41
4.8. SALVARE E LEGGERE UN PROGETTO DI TPA_C.....	41
4.8.1. <i>Codice di esempio</i>	41



1. PRESENTAZIONE

TpaCutOEM.dll (alias TPA_C) è un prodotto realizzato per gli sviluppatori di software e può essere integrato come libreria per una architettura Windows a 32 e 64-bits.

TPA_C è una libreria sviluppata per l'ottimizzazione automatica del taglio lineare di forme rettangolari e permette lo sviluppo di applicazioni di ottimizzazione di piazzamenti 2D in vari settori applicativi.

L'utilizzo tipico della libreria è relativo alla programmazione di macchine sezionatrici.

La libreria deve essere integrata come componente del vostro prodotto.

1.1. VERSIONI

Versione	Data di rilascio	Commenti
1.1.0	05/02/2024	

1.2. LICENZA

L'operatività di TPA_C richiede la verifica di chiave hardware.

La chiave permette anche di salvare il risultato dell'ottimizzazione in formato XML.

1.3. PROPRIETÀ E DIRITTI D'AUTORE

Nessuna parte di questo documento può essere riprodotta, modificata, integrata o tradotta senza previa autorizzazione scritta del proprietario dei diritti d'autore.

Le informazioni contenute in questo documento sono soggette a modifiche e non rappresentano un impegno da rettangolo di TPA Srl.

Sebbene sia stato fatto ogni sforzo per garantire l'accuratezza delle informazioni esposte in questo documento, TPA Srl non si assume alcuna responsabilità nei confronti di alcuna parte per eventuali perdite o danni causati da errori od omissioni o da dichiarazioni di qualsiasi tipo in questo documento, i suoi aggiornamenti, i suoi supplementi o edizioni speciali, indipendentemente dal fatto che tali errori siano omissioni o dichiarazioni risultanti da negligenza, incidente o qualsiasi altra causa. TPA Srl, inoltre, non si assume alcuna responsabilità derivante dall'uso delle informazioni qui descritte; né alcuna responsabilità per danni incidentali o consequenziali derivanti dall'uso di questo documento.

Per ulteriori informazioni, contattate:

T.P.A. Srl Tecnologie e Prodotti per l'Automazione - Via Carducci, 221 - 20099 Sesto S. Giovanni (MI) - ITALIA

Tel. +390236527550

e-mail: info@tpaspa.it

oppure visitate il nostro sito web:

www.tpaspa.com

2. GUIDA UTENTE

TPA_C assegna la classe **TpaRctCut** nel namespace **TpaCutOEM**.

TPA_C realizza il calcolo del taglio lineare di forme rettangolari in aree di piazzamento rettangolari.

Un *rettangolo* è una singola entità che può essere realizzata mediante esecuzione di tagli multipli operati su un'area sezionabile.

Le aree sezionabili sono chiamate *fogli* o *lastre* e anch'essi sono sempre di forma rettangolare.

L'assegnazione dei *rettangoli* permette di definire informazioni di carattere generico (quantità disponibile, priorità di piazzamento) o generalmente tecnologico (spessore, materiale, venatura, possibilità di rotazione, ...). Per ogni tipologia di rettangolo si definiscono la quantità richiesta da piazzare ed una eventuale quantità extra piazzabile in più, a riempimento dei fogli già utilizzati.

L'assegnazione dei *fogli* permette di definire impostazioni di carattere generalmente tecnologico (spessore, materiale, venatura, ...). Per ogni tipologia di foglio si definisce la quantità disponibile.

Le informazioni di carattere generalmente tecnologico permettono di applicare alla procedura di ottimizzazione condizioni di corrispondenza e/o di filtri tra *rettangoli* e *fogli*.

Il *risultato* dell'ottimizzazione fornisce all'utente informazioni dettagliate sul modo migliore per tagliare i *rettangoli* dai *fogli disponibili*. Un *risultato* di ottimizzazione può assegnare uno o più *fogli*: ogni *foglio* contiene il taglio di almeno un *rettangolo*.

La figura illustra un esempio di ottimizzazione di un foglio, altrimenti definito *schema di taglio*:



2.1. OTTIMIZZAZIONE DI TAGLIO

I rettangoli indicati sullo schema di taglio di una lastra sono ottenuti con successione di tagli orizzontali e verticali, alternati in nidificazioni successive.

Il risultato dello schema di taglio che è calcolato per una lastra ottiene il taglio delle singole parti mediante una suddivisione alternata in tagli orizzontali e verticali, fino ad ottenere tutti le partizioni richieste.

L'insieme delle procedure di calcolo che porta a una soluzione di taglio è indicata come *fase di ottimizzazione*. Il taglio di un rettangolo può applicare rotazioni di 0° o 90° (in senso antiorario) al rettangolo medesimo.

Una ottimizzazione può proporre fino ad un massimo di 5 soluzioni, valutabili dall'utente per la scelta della soluzione ritenuta migliore.

2.2. UNITÀ E COORDINATE

In TPA_C i valori metrici (coordinate, dimensioni) sono espressi nell'unità definita come [mm] o [inch] nella proprietà [Unit](#) (0= [mm], 1= [inch]).

TPA_C opera in un sistema 2D di coordinate cartesiane:

- il punto di coordinate [0;0] è posizionato sull'angolo inferiore sinistro del foglio
- X è l'asse orizzontale, positivo verso destra
- Y è l'asse verticale, positivo verso l'alto.

Le coordinate dei tagli che sono calcolati si riferiscono sempre all'angolo in basso a sinistra del foglio.

2.3. DIREZIONE E VERTICE DI SVILUPPO

La Direzione assegna la direzione per il primo taglio di un foglio:

- *direzione orizzontale*: i tagli inizieranno in orizzontale, con livello di taglio iniziale pari a 1
- *direzione verticale*: i tagli inizieranno in verticale, con livello di taglio iniziale pari a 0 (altrimenti chiamati: pre-tagli)
- *direzione non settata*: ogni singolo foglio dell'ottimizzazione potrebbe avere direzione verticale od orizzontale, in base a quale produce il risultato migliore per quel foglio.

La selezione del Vertice assegna lo spigolo di partenza per i piazzamenti sul foglio, con gli sfridi accumulati intorno allo spigolo opposto. I valori sono quattro:

- 0 = Sinistro-Inferiore
- 1 = Sinistro-Superiore
- 2 = Destro-Inferiore
- 3 = Destro-Superiore

TPA_C utilizza sempre l'impostazione del Vertice come assegnata.

Le informazioni corrispondono alle assegnazioni generali di progetto [Direction](#) e [Corner](#).

2.4. GRUPPI E FILTRI DI CORRISPONDENZA

L'assegnazione dei *rettangoli* e dei *fogli* prevede impostazioni di carattere generalmente tecnologico, allo scopo di applicare condizioni di corrispondenza e/o di filtri tra *rettangoli* e *fogli*.

Esaminiamo dapprima i settaggi che concorrono ad applicare condizioni di corrispondenza.

I settaggi interessati corrispondono a campi in strutture [OneRect](#) e [OneSheet](#):

- *Thickness*: (tipo: double) spessore
- *Material*: (tipo: stringa) assegnazione generica di materiale
- *Grain*: (tipo: intero) assegnazione direzione di venatura o grana.

Campo Thickness: la corrispondenza è valutata sull'uguaglianza dei valori impostati, a meno di un epsilon di confronto lineare corrispondente a 0.1 mm.

Esempi di assegnazione di gruppi di corrispondenza sullo spessore di rettangoli e fogli:

- sono assegnati due rettangoli (di identificativi: 1, 2) con campo *Thickness* =18.0
- è assegnato un rettangolo (di identificativo: 3) con campo *Thickness* =25.0
- è assegnato un foglio (di identificativo: 1) con campo *Thickness* =0.0
- è assegnato un foglio (di identificativo: 2) con campo *Thickness* =18.0

la situazione determina l'assegnazione di 3 gruppi di corrispondenza:

- gruppo 1, in corrispondenza a *Thickness* =18.0
- gruppo 2, in corrispondenza a *Thickness* =25.0
- gruppo 3, in corrispondenza a *Thickness* =0.0.

Ogni gruppo è ottimizzato in modo separato. Dall'esempio sopra esposto, è evidente come il solo gruppo 1 possa determinare una soluzione di taglio, permettendo l'associazione tra rettangoli e fogli: i rettangoli di identificativi (1, 2) potranno essere piazzate in fogli di identificativo 2.

In caso di assegnazione di valori multipli (ad esempio: anche per il materiale) le associazioni di corrispondenza tra rettangoli e fogli possono aumentare di complessità. Un esempio:

- gruppo 1, in corrispondenza a *Thickness* = 18.0 e *Material*= "mogano"
- gruppo 2, in corrispondenza a *Thickness* = 18.0 e *Material*= "betulla"
- gruppo 3, in corrispondenza a *Thickness* = 25.0 e *Material*= "mogano"

2.4.1. IL CONTROLLO DELLA VENATURA

Il campo compare in strutture [OneRect](#) e [OneSheet](#), in corrispondenza all'assegnazione della direzione di venatura o grana:

- l'assegnazione avviene in campo *Grain* delle strutture
- il significato tecnologico dell'informazione dipende dal materiale delle lastre (esempi: legno, impiallacciato, metallo).

I valori assegnabili sono tre:

- 0 (*None*): non assegna venatura
- 1 (*X*): assegna venatura lungo la direzione orizzontale
- 2 (*Y*): assegna venatura lungo la direzione verticale.

L'assegnazione di venatura non porta necessariamente alla determinazione di gruppi separati di corrispondenza: rettangoli con stesso valore in campo *Grain* possono essere piazzate in lastre di gruppi differenti e/o con stesso o differente valore di campo *Grain*.

Vediamo quali criteri sono applicati, se è permessa la rotazione al rettangolo:

- un rettangolo con *Grain*=(1, 2) può essere piazzato con rotazione qualunque in foglio con *Grain*=0
- un rettangolo con *Grain*=(1, 2) può essere piazzato in foglio con *Grain*=(1,2) con rotazione tale da rispettare la direzione assegnata per entrambi
- un rettangolo con *Grain*=0 può essere piazzata con rotazione qualunque indipendentemente dal campo *Grain* dei fogli.

Un rettangolo con *Grain*=(1, 2) ma con esclusa possibilità di rotazione potrà essere piazzato solo in fogli con stesso *Grain*.

Se un rettangolo con rotazione abilitata e con *Grain*=(1, 2) può essere piazzata in lastre differenti per venatura non viene garantito un piazzamento privilegiato in lastra con assegnata stessa venatura.

2.5. I TAGLI

Varie informazioni concorrono a determinare il posizionamento e la successione dei tagli, in relazione sia ai rettangoli che devono essere tagliati che ai bordi del foglio.

Un'informazione generale assegna il diametro dell'utensile utilizzato per i tagli: corrisponde al settaggio [CutterDiameter](#).

L'utensile è utilizzato per tutti i tagli previsti dallo schema di taglio.

Le tipologie di taglio previste sono due:

- **Longitudinali**: il taglio percorre il pannello lungo la dimensione X
- **Trasversali**: il taglio percorre il pannello lungo la dimensione Y.

I tagli che sono assegnati in uno schema di taglio vengono differenziati sulla base di codici specifici, di seguito indicati come *livelli di taglio*:

- **Pre-taglio**: taglio di tipo trasversale che genera un *pannello* (livello 0): un pre-taglio può derivare solo da ottimizzazione di un foglio con *Direzione verticale*
- **Longitudinale**: taglio di tipo longitudinale che genera una *striscia* sull'intera lunghezza del pannello (livello 1)
- **Trasversale**: taglio di tipo trasversale che genera un elemento sezionando una striscia (livello 2)
- **Zeta**: taglio di tipo longitudinale che genera un elemento sezionando una porzione di pannello ottenuta con taglio di livello 2 (livello 3)
- **W**: taglio di tipo trasversale che genera un elemento sezionando una porzione di pannello ottenuta con taglio di livello 3 (livello 4)
- **Taglio #5**: taglio di tipo longitudinale che genera un elemento sezionando una porzione di pannello ottenuta con taglio di livello 4 (livello 5)
- **Taglio #6**: taglio di tipo trasversale che genera un elemento sezionando una porzione di pannello ottenuta con taglio di livello 5 (livello 6).

La sequenza dei livelli corrisponde al numero di volte in cui un materiale di base deve essere girato dal taglio longitudinale al taglio trasversale e viceversa per completare un pezzo. Ogni giro di un pannello o di una striscia aumenterà il livello di uno. Questo può essere illustrato come segue, poniamo per il caso di *Direzione orizzontale*:

- la prima direzione di taglio è il livello 1 (longitudinale)
- le strisce di materiale ottenute dovranno essere tagliate nell'altra direzione (livello 2, trasversale)
- se necessario, una singola porzione così ottenuta può essere suddivisa con ulteriori tagli longitudinali (livello 3)
- e così via, fino ad un massimo di 6 livelli di taglio.

2.5.1. LIVELLI DI TAGLI

La sequenza dei tagli di uno schema ottenuto dall'ottimizzazione rispetta un livello massimo di tagli, come assegnato in proprietà [MaxCutLevels](#):

- **Livello 1**: in uno schema sono inseriti solo pre-tagli (se ottimizzazione in *Direzione verticale*) o tagli longitudinali (se ottimizzazione in *Direzione orizzontale*)
- **Livello 2**: in una striscia sono inseriti solo tagli trasversali
- **Livello 3**: in un elemento trasversale sono inseriti solo tagli Z
- **Livello 4**: in un elemento zeta sono inseriti solo tagli W
- **Livello 5**: in un elemento W sono presenti solo tagli di livello 5
- **Livello 6**: nell'ottimizzatore non vengono inseriti tagli oltre il livello 6

Il massimo dei livelli di taglio è pari a 6. I tagli di livello 5 e 6 non hanno un nome specifico.

Il numero di livelli consentiti determinerà la complessità degli schemi di taglio:

- una limitazione del numero di livelli semplificherà gli schemi di taglio
- schemi di taglio semplificati ridurranno i tempi delle operazioni di taglio
- una limitazione del numero di livelli può comportare un maggiore spreco di materiale.

Le figure che seguono corrispondono a differenti valori della proprietà [MaxCutLevels](#):

<i>MaxCutLevels</i>	<i>Direzione= Orizzontale</i>	<i>Direzione= Verticale</i>
---------------------	-------------------------------	-----------------------------





Colori differenti sono utilizzati per rappresentare i vari livelli di taglio:

- **Pre-tagli:** colore verde
- **Tagli longitudinali** (livello 1): colore rosso
- **Tagli trasversali** (livello 2): colore marrone
- **Tagli Z** (livello 3): colore verde oliva
- **Tagli W** (livello 4): colore viola.

2.5.2. RIFILI

È possibile aggiungere distanzamenti in base al livello di taglio usando le seguenti proprietà di tipo *double*:

- *PreCut*: quantità aggiunta a tutti i pre-tagli (tagli di livello 0)
- *LongCut*: quantità aggiunta a tutti i tagli di primo livello
- *TransvCut*: quantità aggiunta a tutti i tagli di secondo livello
- *ZCut*: quantità aggiunta a tutti i tagli di terzo livello.

I valori impostati sono sommati al diametro di taglio.

Ad esempio: con (CutterDiameter = 5, LongCut = 8), tutti i tagli di primo livello hanno spessore 13.

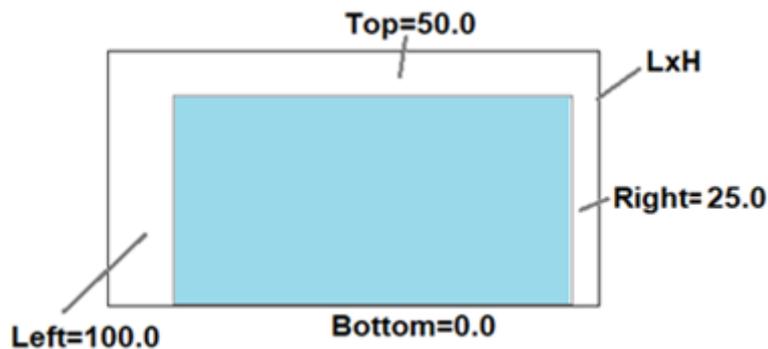
2.5.3. TAGLIO DI RILASCIO DELLA TENSIONE

La funzionalità indicata come TFC (Tension Free Cut) permette di aggiungere ulteriori tagli in corrispondenza dei tagli longitudinali, in modo da eliminare l'effetto "banana" dovuto a tensioni presenti nella lastra.

L'assegnazione corrisponde alla proprietà di tipo *double* [TensionGap](#): il valore è sommato sia al diametro di taglio che al parametro *LongCut*.

2.5.4. MARGINI DEL FOGLIO

Date le dimensioni di un *foglio*, è possibile assegnare delle aree laterali non utili ai fini dei piazzamenti, differenziabili per lato. In figura un foglio di dimensioni LxH e con assegnazione dei quattro margini esterni tutti differenti: l'area utile per i piazzamenti è quella interna colorata.



L'impostazione dei margini esterni corrisponde a campi di struttura [OneSheet](#): BorderLeft, BorderRight, BorderTop e BorderBottom.
Per tutti i campi è possibile assegnare valore nullo o positivo.

2.6. L'OTTIMIZZAZIONE IN TPA_C

L'ottimizzazione in TPA_C ha lo scopo primario di ottenere il migliore utilizzo complessivo del materiale disponibile (*lastre*) con i piazzamenti richiesti (*rettangoli*) e con le condizioni poste (*settaggi generali*).

L'ottimizzazione esegue essenzialmente un numero di *tentativi* e sceglie il risultato del tentativo che offre il miglior utilizzo e, quindi, la *soluzione migliore*.

La sequenza dei vari tentativi modifica alcuni *parametri* in modo da generare situazioni di piazzamenti differenti.

La natura di questi che definiamo generalmente *parametri* riguarda:

- modifica della logica di riempimento di un foglio
- modifiche dell'ordine di piazzamento dei rettangoli
- modifica della rotazione di un rettangolo
- modifica sulla direzione di taglio (orizzontale o verticale; solo se abilitato con la proprietà [Direction](#)).

Ma cosa qualifica una situazione migliore di un'altra?

Vediamo alcuni criteri molto generali:

- è privilegiata la maggiore area dei piazzamenti. Una soluzione che dispone pezzi a occupazione del 93.0% delle lastre è migliore di una che determina un riempimento all'88.00%;
- è privilegiata la soluzione "più riutilizzabile" (significa avere un'unica area di sfrido sufficientemente grande da poter essere riutilizzata);
- sono applicati ulteriori criteri di valutazioni relative alla disposizione dei pezzi, oltre che al numero e dimensione dei pezzi piazzati e agli sfridi interni.

La procedura di ottimizzazione porta alla definizione di una soluzione determinabile in modo ripetitivo, mantenendo invariate tutte le impostazioni al contorno che possono condizionarne lo sviluppo.

2.6.1. OTTIMIZZAZIONI PROGRESSIVE

È possibile che l'ottimizzatore calcoli e renda accessibili più soluzioni, fino a un massimo di 5.
Tutte le soluzioni proposte utilizzano lo stesso numero di fogli.

Tutte le ottimizzazioni calcolate rimangono disponibili fino a:

- una nuova ottimizzazione totale (vedi funzione: [Compute](#))
- una richiesta di inizializzazione totale (vedi funzione: [ClearAll](#)).

2.6.2. CRITERI E PRIORITÀ DI OTTIMIZZAZIONE

Su richiesta di ottimizzazione, TPA_C avvia un'analisi delle liste assegnate e la successiva fase di ottimizzazione.

L'analisi delle liste può riscontrare situazioni di errore, con conseguente annullamento dell'ottimizzazione.

Esaminiamo questa prima fase di analisi, distinguendo tra rettangoli e fogli.

Verifiche sulla lista dei *rettangoli*:

- sono esclusi dall'ottimizzazione rettangoli non abilitati (campo *Enable* in struttura *OneRect*)
- sono esclusi dall'ottimizzazione rettangoli con quantità richiesta (campo *N*) ed extra (campo *Extra*) entrambi nulli
- sono esclusi dall'ottimizzazione rettangoli con una o entrambe le dimensioni assegnate < 1.0 mm
- sono esclusi dall'ottimizzazione rettangoli che non hanno alcuna corrispondenza con *fogli* validi (esempio: rettangolo con *Material*="abc" e nessun foglio con stessa impostazione)

Verifiche sulla lista dei *fogli*:

- sono esclusi dall'ottimizzazione fogli non abilitati (campo *Enable* in struttura *OneSheet*)
- sono esclusi dall'ottimizzazione fogli con quantità disponibile (campo *N*) nulla
- sono esclusi dall'ottimizzazione fogli con una o entrambe le dimensioni < 1.0 mm
- sono esclusi dall'ottimizzazione fogli che non hanno alcuna corrispondenza con *rettangoli* validi (esempio: foglio con *Material*= "abc" e nessun rettangolo con stessa impostazione).

Il risultato delle analisi riportate deve potere utilizzare almeno un elemento per ogni lista di rettangoli e lastre.

L'avvio della procedura di ottimizzazione applica criteri specifici per assegnare l'ordine di utilizzo di rettangoli e fogli.

2.6.2.1. UTILIZZO DELLE LASTRE

L'utilizzo delle lastre rispetta un ordine che può tenere in considerazione differenti situazioni.

Vediamo ora i principali criteri che sono applicati all'ordinamento iniziale dei fogli. I punti sono applicati secondo l'ordine riportato, passando al successivo nel caso in cui non è stato possibile effettuare una scelta prevalente sul punto precedente:

- ordinamento per priorità crescente (eccetto 0, che esegue per ultimo)
- ordina prima i fogli marcati come sfrido (campo *IsScrap* in struttura [OneSheet](#)), con priorità uguale
- il settaggio generale [OrderSheetDim](#) abilita l'ordinamento delle lastre per dimensione (ordina per area decrescente)
- ordina per tipo crescente (campo *ID* in struttura [OneSheet](#)) a parità delle altre condizioni ai punti precedenti.

Il settaggio generale [UseBeforeScrap](#) abilita l'applicazione del campo di qualifica dei *fogli* come sfrido.

2.6.2.2. UTILIZZO DEI RETTANGOLI

Vediamo ora i principali criteri che sono applicati all'ordinamento iniziale dei rettangoli. I punti sono applicati secondo l'ordine riportato, passando al successivo nel caso in cui non è stato possibile effettuare una scelta prevalente sul punto precedente:

- ordinamento per priorità crescente (eccetto 0, che esegue per ultimo)
- rettangoli con quantità richiesta assegnata (positiva). Cioè: ordina dopo i rettangoli che richiedono solo piazzamenti extra
- ordina per area decrescente
- rettangolo senza possibilità di rotazione
- rettangolo con quantità richiesta maggiore
- rettangolo con tipo crescente (campo *ID* in struttura [OneRect](#))

I PIAZZAMENTI EXTRA

Per i rettangoli è possibile assegnare piazzamenti extra, in aggiunta o in alternativa a quelli effettivamente richiesti. Per un rettangolo è, cioè, possibile assegnare: anche o solo piazzamenti extra.

L'informazione è assegnata in campo *Extra* di struttura [OneRect](#): un valore positivo assegna direttamente il numero dei piazzamenti extra.

Su un foglio, i piazzamenti extra sono utilizzati solo dopo verifica di non potere più piazzare rettangoli richiesti. Risulta quindi evidente come in nessun caso un foglio possa essere utilizzato solo per applicare piazzamenti extra.

2.7. SUPPORTO PER LA GESTIONE DEI PROGETTI DI TAGLIO

Un progetto di taglio è inteso come l'insieme di tutte le informazioni assegnate alla libreria TPA_C:

- settaggi di abilitazioni complessive
- liste di rettangoli, fogli.

TPA_C espone metodi di serializzazione di un progetto. Questi metodi sono di particolare utilità per:

- situazioni di testing
- casi di integrazione di TPA_C in cui le informazioni qui previste nelle strutture di assegnazione delle liste sono sufficienti.

2.8. LIMITI CONOSCIUTI DELLA LIBRERIA

Assegnazione dei rettangoli

- È possibile assegnare fino a un massimo di 500 differenti *rettangoli*
- per ogni *rettangolo*, è possibile assegnare una quantità massima piazzabile pari a 999.

Assegnazione dei fogli

- è possibile assegnare fino a un massimo di 100 differenti *fogli*
- per ogni *foglio*, è possibile assegnare una quantità massima utilizzabile pari a 999.

3. GUIDA ALLE FUNZIONI DELLA LIBRERIA

Il capitolo descrive in dettaglio proprietà e funzioni di TPA_C.

3.1. GESTIONE DELLA LICENZA

3.1.1. ISVALIDLICENSE

Proprietà di tipo **boolean** che testa presenza e stato della chiave.

Valore

True se il modulo è abilitato, *False* altrimenti.

Note

L'interrogazione della proprietà esegue una effettiva lettura della chiave, ma solo se non è in corso una ottimizzazione di taglio.

Nessuna ottimizzazione può essere eseguita se la chiave non è valida.

La verifica della presenza e dello stato della chiave è eseguita dalla libreria con cadenza temporale.

3.2. COSTANTI

Le costanti seguenti sono disponibili nella classe *TpaCutOEM.TpaRctCut*:

MAX_ROW_ITEMS	500	Numero massimo di tipi di rettangoli
MAX_ROW_N	999	Valore massimo di piazzamenti richiesti per un rettangolo
MAX_SHEET_ITEMS	100	Numero massimo di tipi di fogli
MAX_SHEET_N	999	Quantità massima di piazzamenti disponibile per un foglio
MAX_CUSTOM_SETTINGS	10	Numero massimo di settaggi generali di progetto
MAX_LL_SETTINGS	250	Lunghezza massima di singolo settaggio generale di progetto (lunghezza di stringa)
MAX_ROW_PARAMS	15	Numero massimo di settaggi generali in singolo rettangolo
MAX_LL_PARAMS	50	Lunghezza massima di singolo settaggio generale di rettangolo (lunghezza di stringa)
MAX_CUT_LEVELS	6	Numero Massimo dei livelli di taglio

3.3. ENUMERAZIONI

Le enumerazioni seguenti sono disponibili nel namespace **TpaCutOEM**.

3.3.1. CUTERROR

Assegnazione degli errori gestiti dalla libreria

- *ErrorNone*: nessun errore
- *ErrorKey*: licenza non verificata
- *ErrorMemory*: errore di memoria
- *ErrorRectEmpty*: lista rettangoli vuota o nessun rettangolo abilitato o piazzabile
- *ErrorSheetEmpty*: lista lastre vuota o nessuna lastra abilitata
- *ErrorNoMatch*: nessun match di corrispondenza con la lista delle lastre
- *ErrorIOfile*: errore in gestione di file progetto (errore di accesso a percorso e/o file, ...)
- *ErrorFileNotValid*: errore in gestione di file di progetto (formato non valido)
- *ErrorNoSolutions*: nessuna soluzione risulta assegnata
- *ErrorUnexpected*: errore generale o non gestito (es: indice assegnato non valido).

3.4. STRUTTURE

Le strutture seguenti sono disponibili nel namespace **TpaCutOEM**.

3.4.1. ONERECT

Assegnazione generale di un *rettangolo*. La struttura espone metodi di inizializzazione di campi ai valori di default.

- *int ID*: identificativo numerico del *rettangolo* (univoco, strettamente positivo) *{default = 0}*
- *bool Enable*: abilitazione all'utilizzo del *rettangolo* (false = non piazza il rettangolo) *{default = true}*
- *string Label*: nome descrittivo del rettangolo (può essere =""; lunghezza massima 50 caratteri) *{default = ""}*
- *double Length*: lunghezza (≥ 0.0) *{default = 0.0}*
- *double Height*: altezza (≥ 0.0) *{default = 0.0}*
 - i valori sono in unità di: [Unit](#)
- *double Thickness*: spessore (≥ 0.0) *{default = 0.0}*.

- assegnare il campo con valori differenziati se è necessario applicare una corrispondenza con il corrispondente campo dei *fogli*
- i valori sono in unità di: [Unit](#)
- *int N*: quantità richiesta per piazzamenti (≥ 0) {*default* = 0; valore massimo = 999}
- *int Extra*: quantità extra disponibile (significativa se > 0) {*default* = 0; valore massimo = 999}
 - il campo *N* imposta la quantità richiesta per il rettangolo. Il campo *Extra* imposta la quantità extra. I pezzi extra vengono inseriti solo dopo avere cercato di piazzare le quantità richieste di tutte le tipologie di *rettangoli*.
- *string Material*: materiale {*default* = ""}
 - assegnare il campo con valori differenziati se è necessario applicare una corrispondenza con il corrispondente campo dei *fogli*
 - il reale significato attribuito al campo compete all'applicazione esterna
- *int Grain*: direzione venatura (0 = none; 1 = orizzontale; 2 = verticale) {*default* = 0}
 - assegnare il campo con valore 1 o 2 se è necessario applicare una corrispondenza con il corrispondente campo dei *fogli*. Un rettangolo con campo (1, 2) è piazzabile su *foglio* con venatura solo se è possibile rispettare la direzione della venatura stessa (con eventuale applicazione di rotazione); un rettangolo senza venatura è applicabile a lastre con campo *Grain* qualsiasi
- *int Priority*: priorità di utilizzo (≥ 0) {*default* = 0}
 - *rettangoli* con priorità di numero minore hanno precedenza di piazzamento nella soluzione dello schema di taglio
 - *rettangoli* con priorità 0 verranno inseriti per ultimi
- *bool Rotate*: rotazione (*false* = non ammessa; *true* = ammessa) { *default* = *true*}
 - il valore *true* corrisponde ad abilitare la rotazione di 90° in senso antiorario
- *string MatEdge1, MatEdge2, MatEdge3, MatEdge4*: materiale dei bordi, differenziati sui 4 lati del rettangolo (rispettivamente: superiore, inferiore, destro, sinistro) {*default* = ""}
- *double ThickEdge1, ThickEdge2, ThickEdge3, ThickEdge4*: spessore dei bordi, differenziati sui 4 lati del rettangolo {*default* = 0}
- *bool BoolEdge1, BoolEdge2, BoolEdge3, BoolEdge4*: abilitazione alla riduzione del rettangolo degli spessori corrispondenti ai bordi, differenziati sui 4 lati del rettangolo, da utilizzarsi in ottimizzazione di taglio {*default* = *false*}
 - l'ottimizzatore di taglio utilizza le informazioni dei bordi per applicare una possibile riduzione della dimensione del rettangolo, come ottenuto dallo schema di taglio. La riduzione è applicabile su ogni lato del rettangolo, con verifica di corrispondenti informazioni (*ThickEdge* > 0.0 , *BoolEdge* = *true*)
 - i campi relativi ai bordi saranno utilizzabili in stazione bordatrice, a valle di una ottimizzazione di taglio
 - i campi relativi ai bordi sono significativi solo nel caso in cui risulta [EnableRectEdge](#)=*true*.
- *string Param1, Param2..., Param15*: 15 parametri generici che non influenzano l'ottimizzazione {*default* = ""}
 - i campi sono disponibili per assegnazione di informazioni di carattere gestionale o generalmente tecnologico. Esempi: informazioni relative alla commessa (cliente, numero d'ordine, annotazioni), informazioni aggiunte relative al materiale.

3.4.2. ONESHEET

Assegnazione generale di un *foglio*. La struttura espone metodi di inizializzazione di campi ai valori di default.

- *int ID*: identificativo numerico del *foglio* ([univoco](#), strettamente positivo) {*default* = 0}
- *bool Enable*: abilitazione all'utilizzo del *foglio* (*false* = non piazza il foglio) {*default* = *true*}
- *string Label*: nome descrittivo del *foglio* (può essere =""; lunghezza massima 50 caratteri) {*default* = ""}
- *bool Scrap*: identifica il foglio come recuperato (Es.: sfrido di lastra ottimizzata in precedenza) {*default* = *false*}
 - *fogli* di questo tipo possono avere precedenza di utilizzo
 - l'applicazione del campo è assegnata dal settaggio [OrderBeforeScrap](#)

- *double Length*: lunghezza (≥ 0.0) {*default* = 0.0}
- *double Height*: altezza (≥ 0.0) {*default* = 0.0}
 - i valori sono in unità di: [Unit](#)
- *double Thickness*: spessore (≥ 0.0) {*default* = 0.0}.
 - assegnare il campo con valori differenziati se è necessario applicare una corrispondenza con il corrispondente campo dei *rettangoli*
 - i valori sono in unità di: [Unit](#)
- *int N*: quantità disponibile (≥ 0) {*default* = 0; valore massimo = 100}
- *string Material*: materiale {*default* = ""}
 - assegnare il campo con valori differenziati se è necessario applicare una corrispondenza con il corrispondente campo dei rettangoli
 - il reale significato attribuito al campo compete all'applicazione esterna
- *int Grain*: direzione venatura (0 = none; 1 = x = orizzontale; 2 = y = verticale) {*default* = 0}
 - assegnare il campo con valore 1 o 2 se è necessario applicare una corrispondenza con il corrispondente campo dei *rettangoli*
- *int Priority*: priorità di utilizzo (≥ 0) {*default* = 0}
 - *fogli* con priorità di numero minore hanno precedenza di utilizzo
 - *fogli con priorità = 0* saranno utilizzati per ultimi
- *double Cost*: costo del foglio per unità di superficie {*default* = 0}
 - l'unità dell'informazione è assegnata dal client, come pure ne è delegato al client l'utilizzo
 - unità tipiche possono essere: €/mq (euro/metro quadrato), \$/ft2 (dollaro/piede quadrato)
- *double BorderBottom, BorderTop, BorderRight, BorderLeft*: margini del foglio {*default* = 0}
 - i valori sono in unità di: [Unit](#)

3.5. DEFINIZIONE DELLE FUNZIONALITÀ

Esaminiamo qui come assegnare tutte le funzionalità di TPA_C: in riferimento alle funzionalità si utilizza anche la dicitura di settaggi.

In fase di inizializzazione di TPA_C, tutti i settaggi sono assegnati ai valori di default, di seguito indicati con locuzione {*default* = ...}.

Al fine di fornire un quadro più organico, le assegnazioni sono suddivise in quattro raggruppamenti:

- *Funzionalità generali*: assegnazioni di utilizzo generico
- *Funzionalità corrispondenti a settaggi generali di ottimizzazione di taglio*: assegnazioni che hanno significato di personalizzazioni generali dei progetti e dell'ottimizzazione di taglio. Si tratta di informazioni che generalmente rimangono invariate e non sono serializzate in file corrispondente ad un progetto
- *Funzionalità relative alle assegnazioni generali di un progetto di taglio*: assegnazioni che hanno significato di assegnazioni specifiche di un progetto. Si tratta di informazioni che sono generalmente serializzate in file corrispondente ad un progetto

3.5.1. FUNZIONALITÀ GENERALI

3.5.1.1. VERSION

Proprietà di tipo **string** che restituisce la versione della libreria.

La stringa riporta il numero principale, secondario, di build e di revisione della libreria.

3.5.1.2. LASTERROR

Proprietà di tipo **CutError** che restituisce l'ultimo errore riscontrato. Il valore è aggiornato in tutte le procedure che possono diagnosticare una situazione anomala.

3.5.1.3. ERRORMESSAGE

La funzione restituisce il messaggio assegnato per l'errore indicato.

string ErrorMessage (CutError Error)

Argomenti

- *Error*: valore di enumerazione

Valore di ritorno

Se è *Error = CutError.ErrorNone*, il ritorno della funzione è "" (stringa vuota).

I messaggi interni sono assegnati in lingua inglese.

3.5.2. FUNZIONALITÀ CORRISPONDENTI A SETTAGGI GENERALI DI OTTIMIZZAZIONE DI TAGLIO

Si tratta di informazioni che non cambiano al variare del progetto: riguardano impostazioni generalmente attribuibili ad opzioni di funzionamento generale di una applicazione.

3.5.2.1. NUMBERCUSTOM

Proprietà di tipo **intero** che assegna/restituisce il numero di settaggi generali del progetto *{default = MAX_CUSTOM_SETTINGS, min=0, max = MAX_CUSTOM_SETTINGS }*.

Il valore può limitare il numero dei settaggi generali di progetto che TPA_C gestisce (ad esempio: in lettura e/o registrazione del progetto da e su file).

3.5.2.2. NUMBERRECTPARAM

Proprietà di tipo **intero** che assegna/restituisce il numero di parametri presenti in ciascun OneRect *{default = MAX_ROW_PARAMS, min=0, max = MAX_ROW_PARAMS}*.

Il valore può limitare il numero dei settaggi generali di progetto che TPA_C gestisce (ad esempio: in lettura e/o registrazione del progetto da e su file).

3.5.2.3. ENBLERECTEDGE

Proprietà di tipo **boolean** che assegna/restituisce l'abilitazione all'assegnazione dei bordi dei rettangoli *{default = True}*.

In particolare:

- *true*: l'ottimizzatore potrà utilizzare un rettangolo con dimensioni ridotte, per effetto dell'applicazione dei bordi
- *false*: i campi relativi ai bordi di un rettangolo non sono serializzati (su file di progetto).

3.5.2.4. ORDERSHEETDIM

Proprietà di tipo **boolean** che assegna/restituisce l'abilitazione all'ordinamento delle lastre sulla base delle dimensioni *{default = False}*.

Valore =*true*: abilita l'ottimizzazione ad applicare un ordinamento delle lastre per area decrescente.

3.5.2.5. ORDERBEFORESCRAP

Proprietà di tipo **boolean** che assegna/restituisce l'abilitazione ad utilizzare prima le lastre marcate di sfrido *{default = False}*.

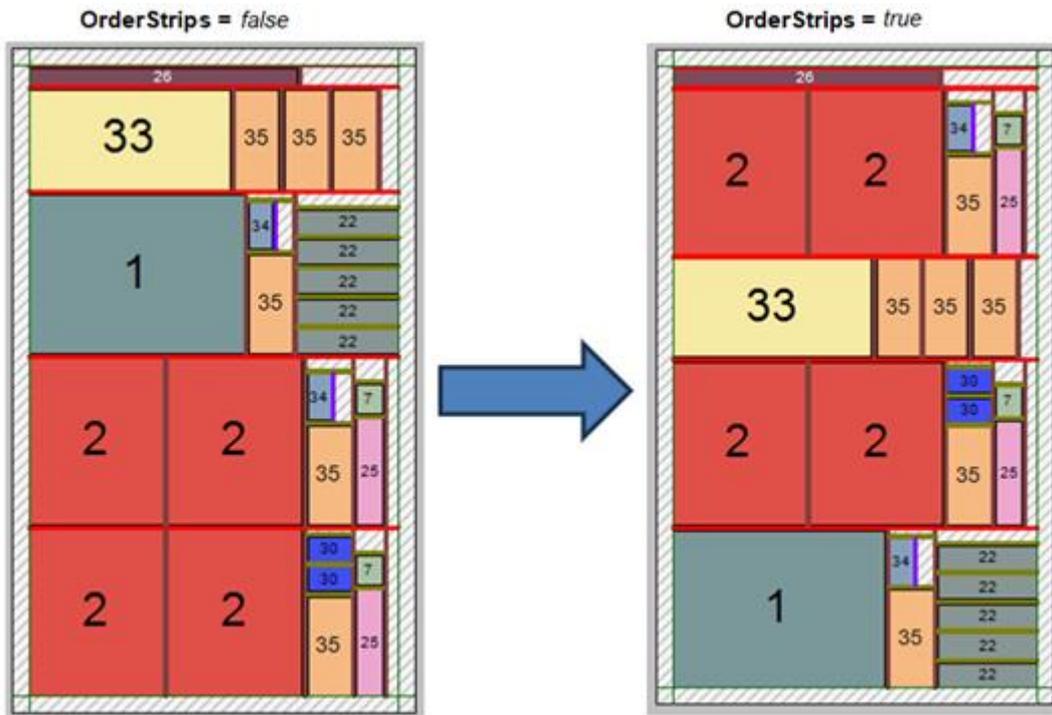
Valore =*true*: abilita l'ottimizzazione ad utilizzare le lastre di sfrido prima delle altre.

3.5.2.6. ORDERSTRIPS

Proprietà di tipo **boolean** che assegna/restituisce l'abilitazione all'ordinamento delle strisce in uno schema di taglio *{default = true}*.

Si rammenta che una striscia corrisponde ad una porzione di pannello ottenuta con taglio longitudinale di livello 1.

La figura riporta lo stesso schema di taglio come ottenuto nei due casi:

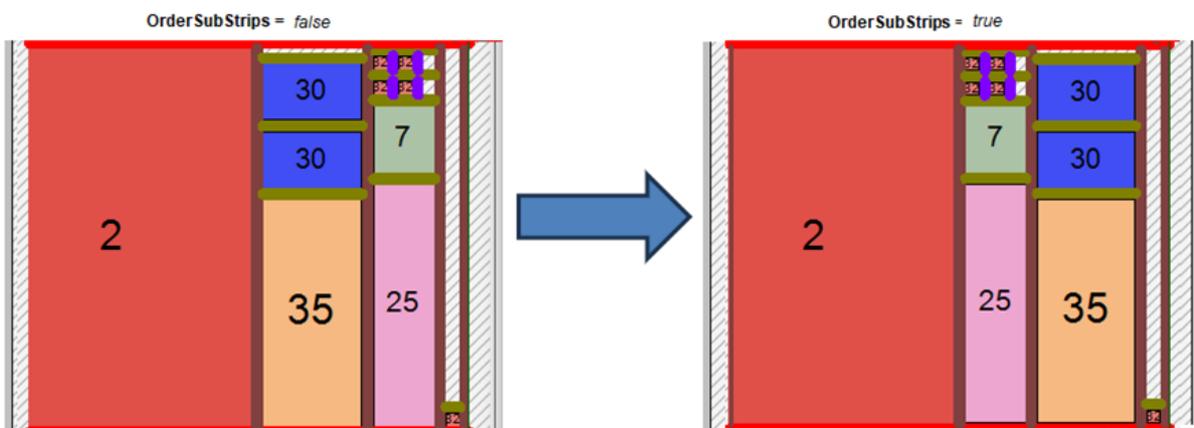


- *false*: (a sinistra) le strisce sono riportate in ordine di calcolo
- *true*: (a destra) l'ordine delle strisce è tale da rispettare
 - o valori crescenti di materiale sprecato all'interno della striscia
 - o valori crescenti dell'altezza delle strisce (a parità di spreco).

3.5.2.7. ORDERSUBSTRIPS

Proprietà di tipo **boolean** che assegna/restituisce l'abilitazione all'ordinamento dei pezzi all'interno di una striscia o porzione di striscia di uno schema di taglio `{default = true}`.

La figura riporta la stessa striscia come ottenuto nei due casi:



- *false*: (a sinistra) i tagli della striscia sono riportati in ordine di calcolo
- *true*: (a destra) l'ordine in ogni porzione della striscia è tale da rispettare
 - o valori crescenti di materiale sprecato all'interno della porzione.

3.5.3. FUNZIONALITÀ RELATIVE ALLE ASSEGNAZIONI GENERALI DI UN PROGETTO

Si tratta di informazioni che possono cambiare al variare del progetto.

3.5.3.1. UNIT

Proprietà di tipo **intero** che restituisce l'unità assegnata con funzione $\{default = 0\}$.

3.5.3.2. CUTTERDIAMETER

Proprietà di tipo **double** che assegna/restituisce il diametro dell'utensile di taglio $\{unità: [Unit](#); default = 0.0; range di validità: >= 0.0\}$.

Il piazzamento tra *rettangoli* applica un distanziamento minimo corrispondente al valore impostato.

3.5.3.3. DIRECTION

Proprietà di tipo **short** che assegna/restituisce la direzione del primo livello di taglio applicato ai fogli $\{default = 0; range di validità: 0/1/2\}$:

- 0 = direzione orizzontale
- 1 = direzione verticale
- 2 = direzione determinata dall'ottimizzatore.

In caso di valore 2, l'ottimizzatore cercherà sempre lo schema di taglio ottimale per ogni tipo di foglio. In questo caso: come risultato di una ottimizzazione, alcuni fogli potrebbero avere la prima direzione di taglio che corrisponde alla dimensione della lunghezza ed altri potrebbero avere la prima direzione di taglio che corrisponde alla dimensione dell'altezza.

3.5.3.4. CORNER

Proprietà di tipo **short** che assegna/restituisce il vertice di partenza per i piazzamenti, tra i valori validi $\{default = 0; range di validità: 0/3\}$

- 0 = Sinistro-Inferiore
- 1 = Sinistro-Superiore
- 2 = Destro-Inferiore
- 3 = Destro-Superiore.

3.5.3.5. MAXCUTLEVELS

Proprietà di tipo **intero** che assegna/restituisce il massimo livello di taglio utilizzabile in ottimizzazione $\{default = MAX_CUT_LEVELS; min = 1; max = MAX_CUT_LEVELS\}$.

Valore 1 corrisponde alla possibilità di eseguire un solo livello di taglio:

- Livello 0 (pre-taglio) in caso di ottimizzazione in direzione verticale
 - richiede rettangoli con altezza uguale a quella del pannello grezzo
- Livello 1 (taglio longitudinale) in caso di ottimizzazione in direzione orizzontale
 - richiede rettangoli con lunghezza uguale a quella del pannello grezzo.

3.5.3.6. TENSIONGAP

Proprietà di tipo **double** che assegna/restituisce una distanza aggiuntiva tra i tagli di primo livello allo scopo di alleviare la tensione della lastra $\{unità: [Unit](#); default = 0.0; range di validità: >= 0.0\}$.

3.5.3.7. PRECUT

Proprietà di tipo **double** che assegna/restituisce un rifilo aggiunto a tutti i pre-tagli (tagli di livello 0) $\{unità: [Unit](#); default = 0.0; range di validità: >= 0.0\}$.

3.5.3.8. LONGCUT

Proprietà di tipo **double** che assegna/restituisce un rifilo aggiunto a tutti i tagli di primo livello $\{unità: [Unit](#); default = 0.0; range di validità: >= 0.0\}$.

3.5.3.9. TRANSVCUT

Proprietà di tipo **double** che assegna/restituisce un rifilo aggiunto a tutti i tagli di secondo livello {unità: [Unit](#); default = 0.0; range di validità: >= 0.0}.

3.5.3.10. ZCUT

Proprietà di tipo **double** che assegna/restituisce un rifilo aggiunto a tutti i tagli di terzo livello {unità: [Unit](#); default = 0.0; range di validità: >= 0.0}.

3.5.3.11. CUSTOM1, CUSTOM2, ..., CUSTOM10

Proprietà di tipo **string** che assegnano/restituiscono informazioni generiche di progetto che non influiscono sul processo di ottimizzazione {default = ""; numero massimo di proprietà = [NumberCustom](#); lunghezza massima di un parametro = MAX_LL_SETTINGS}.

3.5.3.12. CLEARALL

La funzione azzerà le liste del progetto (rettangoli, lastre) ed ogni soluzione calcolata.

bool ClearAll ()

Valore di ritorno

true se l'esito è positivo.

Al momento non sono possibili situazioni che determinano ritorno *false*.

3.6. DEFINIZIONE DEI RETTANGOLI

3.6.1. ADDRCT

La funzione aggiunge un rettangolo alla lista.

bool AddRct (OneRect Item)

Argomenti

- *Item*: struttura di assegnazione del rettangolo

Valore di ritorno

true se l'esito è positivo, *false* se il rettangolo non è stato inserito

Note

Un ritorno *false* corrisponde a una delle situazioni di errore:

- la lista dei rettangoli è già al massimo consentito (500 elementi)
- per il rettangolo è assegnato un identificativo non valido (**Item.ID** deve essere strettamente positivo)
- risulta già assegnata un rettangolo con identificativo numerico **Item.ID**
- il rettangolo presenta dimensioni non valide, o dimensioni dei bordi non valide. In particolare: la sottrazione dello spessore dei bordi alle dimensioni del rettangolo ridurrebbe una o entrambe le dimensioni a valore < 1.0 mm

3.6.2. CLEARRCT

La funzione svuota la lista dei rettangoli.

bool ClearRct ()

Valore di ritorno

True se la lista è stata cancellata correttamente

3.6.3. COUNTRECT

La proprietà di tipo **int** ottiene il numero dei rettangoli in lista.

3.6.4. READRECT

La funzione cerca un rettangolo in corrispondenza all'ID specificato.

CutError ReadRect (int ItemID, ref OneRect Item)

Argomenti

- *ItemID*: identificativo del rettangolo (> 0)
- *Item*: struttura di assegnazione del rettangolo

Valore di ritorno

CutError.ErrorNone se l'esito è positivo

Note

Un ritorno *diverso da CutError.ErrorNone* corrisponde a una delle situazioni di errore:

- (*CutError.ErrorUnexpected*) è assegnato un identificativo non valido (***ItemID*** deve essere strettamente positivo) oppure non risulta assegnato un rettangolo con identificativo numerico ***ItemID***.

3.6.5. READRECTINDEX

La funzione cerca un rettangolo in corrispondenza dell'indice specificato.

CutError ReadRectIndex (int Index, ref OneRect Item)

Argomenti

- *Index*: indice (base zero) sulla lista dei rettangoli (>= 0)
- *Item*: struttura di assegnazione del rettangolo

Valore di ritorno

CutError.ErrorNone se l'esito è positivo

Note

L'utilizzo primario della funzione è per acquisizione dei rettangoli dopo l'esecuzione di funzione ***LoadProject***.

Un ritorno *diverso da CutError.ErrorNone* corrisponde a una delle situazioni di errore:

- (*CutError.ErrorUnexpected*) è assegnato un indice non valido.

3.7. DEFINIZIONE DEI FOGLI

3.7.1. ADDSHEET

La funzione aggiunge un foglio alla lista.

bool AddSheet (OneSheet Item)

Argomenti

- *Item*: struttura di assegnazione del foglio

Valore di ritorno

true se l'esito è positivo

Note

Un ritorno diverso da *true* corrisponde a una delle situazioni di errore:

- La lista dei fogli è già al massimo consentito (100 elementi) o la quantità richiesta in struttura eccede il massimo consentito (999)
- Per il foglio è assegnato un identificativo non valido (***Item.ID*** deve essere strettamente positivo)
- Risulta già assegnato un foglio con identificativo numerico ***Item.ID***

- Il foglio assegna margini non validi. In particolare: la sottrazione dei margini alle dimensioni del foglio riduce una o entrambe le dimensioni a valore < 1.0 mm

3.7.2. CLEARSHEET

La funzione svuota la lista dei fogli.

bool ClearSheet ()

Valore di ritorno

true se l'esito è positivo

3.7.3. COUNTSHEET

La proprietà di tipo **int** ottiene il numero dei fogli in lista.

3.7.4. READSHEET

La funzione cerca un foglio in corrispondenza all'ID specificato.

CutError ReadSheet (int ItemID, ref OneSheet Item)

Argomenti

- *ItemID*: identificativo del foglio (> 0)
- *Item*: struttura di assegnazione del foglio

Valore di ritorno

CutError.ErrorNone se l'esito è positivo.

Note

Un ritorno diverso da *CutError.ErrorNone* corrisponde ad una delle situazioni di errore:

- (*CutError.ErrorUnexpected*) è assegnato un identificativo non valido (**ItemID** deve essere strettamente positivo) oppure non risulta assegnata un foglio con identificativo numerico **ItemID**

3.7.5. READSHEETINDEX

La funzione cerca un foglio in corrispondenza all'indice specificato.

CutError ReadSheetIndex (int Index, ref OneSheet Item)

Argomenti

- *Index*: indice (base zero) sulla lista dei fogli (>= 0)
- *Item*: struttura di assegnazione del foglio

Valore di ritorno

CutError.ErrorNone se l'esito è positivo.

Note

L'utilizzo primario della funzione è per acquisizione dei fogli dopo l'esecuzione di funzione **LoadProject**.

Un ritorno diverso da *CutError.ErrorNone* corrisponde ad una delle situazioni di errore:

- (*CutError.ErrorUnexpected*) è assegnato un indice non valido.

3.8. SOLUZIONE DI TAGLIO

3.8.1. COMPUTE

La funzione avvia la procedura di ottimizzazione

bool Compute ()

Valore di ritorno

true se l'esito è positivo.

Note

In caso di ritorno *false*, si può interrogare la proprietà *LastError* per vedere cosa ha impedito l'ottimizzazione:

- (*CutError.ErrorKey*) si è verificato esito negativo nella verifica di presenza e stato della chiave
- (*CutError.ErrorRectEmpty*) la lista dei rettangoli è vuota oppure non assegna elementi abilitati
- (*CutError.ErrorSheetsEmpty*) la lista dei fogli è vuota oppure non assegna elementi abilitati
- (*CutError.ErrorNoMatch*) la lista delle parti e fogli non hanno gruppi di corrispondenza in comune

All'avvio, la funzione:

- effettua le verifiche preliminari (di cui sopra)
- effettua verifiche sulle liste di rettangoli e fogli.

Se l'esecuzione delle verifiche elencate porta ad una situazione di errore, comunque la funzione ritorna senza eseguire alcuna ottimizzazione.

3.9. I RISULTATI DELL'OTTIMIZZAZIONE

Tutte le proprietà e funzioni qui elencate falliscono se non risulta calcolata una ottimizzazione valida.

3.9.1. NUMBEROFSOLUTIONS

Proprietà di tipo **intero** che restituisce il numero totale delle soluzioni calcolate.

Il valore restituito è:

- 0: nessuna soluzione risulta calcolata
- >0: numero delle soluzioni che risultano calcolate

3.9.2. SELECTSOLUTION

Proprietà di tipo **intero** che assegna/restituisce il numero della *soluzione corrente*.

Il valore restituito è:

- -1: nessuna soluzione è calcolata
- >=0: la soluzione corrente corrisponde ad una calcolata con chiamata di funzione **Compute()**

In assegnazione:

- la proprietà permette di confermare o cambiare la *soluzione corrente*
- non è effettuata assegnazione se il valore non è valido.

3.9.3. SOLUTIONSHEETS

Proprietà di tipo **intero** che restituisce il numero di fogli della *soluzione corrente*.

Il valore restituito è:

- 0: la soluzione è vuota o inesistente
- >0: la soluzione risulta valida

Un foglio di una soluzione è indicato anche con la dicitura: *schema di taglio*.

Il valore restituito non considera le eventuali ripetizioni di fogli.

3.9.4. GETSHEETID

La funzione restituisce l'ID di un foglio dato il suo indice nella *soluzione corrente*.

int GetSheetID (int idx)

Argomenti

- *idx*: indice del foglio (>= 0)

Note

La funzione ritorna 0 in caso di indice o soluzione non validi.

3.9.5. NUMBEROFREPETITIONS

La funzione restituisce quante volte è ripetuto un determinato schema di taglio nella *soluzione corrente*.

int NumberOfRepetitions (int idx)

Argomenti

- *idx*: indice del foglio (>= 0)

Note

Il valore della proprietà è significativo dopo l'esecuzione di una ottimizzazione.

La funzione ritorna 0 in caso di indice o soluzione non valida.

La funzione ritorna 1 se lo schema di taglio è assegnato una volta sola (senza ripetizioni).

3.9.6. USEDSHEETS

La funzione legge il numero dei fogli della *soluzione corrente* (fogli con piazzamenti).

int UsedSheets (int ID_Sheet)

Argomenti

- *ID_Sheet*: identificativo del foglio (> 0)

Note

La funzione torna il numero dei fogli della soluzione corrente o di tipo di foglio:

<i>ID_Sheet</i>	
=0	ottiene il numero complessivo dei fogli calcolati per la soluzione
>0	ottiene il numero dei fogli di tipo (<i>ID_Sheet</i>) calcolati per la soluzione

3.9.7. USEDRECTS

La funzione legge il numero di piazzamenti in un foglio della soluzione corrente o sull'intera soluzione corrente, con possibilità di corrispondenza a un solo tipo di rettangolo.

int UsedRects (int Index, int ID_Part)

Argomenti

- *Index*: indice a base zero del foglio della soluzione
- *ID_Part*: identificativo del rettangolo (> 0)

Note

La funzione legge il numero di piazzamenti corrispondenti agli argomenti assegnati:

<i>Index</i>	<i>ID_Part</i>	
-1	0	ottiene il numero totale dei piazzamenti della soluzione
-1	>0	ottiene il numero dei piazzamenti corrispondenti al rettangolo (<i>ID_Part</i>) in tutti i fogli della soluzione

≥ 0	0	ottiene il numero totale dei piazzamenti in foglio di indice (<i>Index</i>) della soluzione
≥ 0	> 0	ottiene il numero totale dei piazzamenti corrispondenti al rettangolo (<i>ID_Part</i>) in foglio di indice (<i>Index</i>)

Nei casi di funzionamento con *Index* negativo (-1), il valore restituito tiene conto di fogli ripetuti uguali. Nei casi di funzionamento con *Index* ≥ 0 , il valore restituito conteggia i piazzamenti di un singolo foglio, senza considerare le eventuali ripetizioni del foglio stesso.

3.9.8. FITNESSSHEET

La funzione restituisce l'informazione di efficienza di un foglio della *soluzione corrente*.

double FitnessSheet (int idx)

Argomenti

- *idx*: indice del foglio (≥ 0)

Note

Il valore della proprietà è significativo dopo l'esecuzione di una ottimizzazione. L'efficienza complessiva è valutata in % del rapporto tra l'area utilizzata per i piazzamenti e l'area totale dei fogli utilizzati. Se la soluzione genera più fogli, la valutazione di efficienza dell'ultimo foglio può opportunamente limitare l'area utile per i piazzamenti

La funzione torna il fitness in percentuale di un foglio o di tutta la soluzione corrente:

<i>idx</i>	
= -1	ottiene il fitness di tutta la soluzione (prende in considerazione le ripetizioni)
≥ 0	ottiene il fitness del foglio di indice <i>idx</i>

3.9.9. READRESOLRECT

La funzione restituisce le informazioni relative ad un piazzamento su un foglio della *soluzione corrente*

int ReadResolRect (int idx, int idxrct, ref double qx, ref double qy, ref bool rotate)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione
- *idxrct*: indice (base zero) del piazzamento sul foglio della soluzione
- *qx*: coordinata x dell'angolo in basso a sinistra del rettangolo
- *qy*: coordinata y dell'angolo in basso a sinistra del rettangolo
- *rotate*: indica se il rettangolo è ruotato (*true* = ruotato)

Valore di ritorno

L' identificativo del rettangolo di cui si sono letti i dati in caso di esito positive, 0 altrimenti.

Note

In caso di ritorno 0, è possibile interrogare la proprietà *LastError* per sapere l'errore:

- (*CutError.ErrorKey*) si è verificato esito negativo nella verifica di presenza e stato della chiave
- (*CutError.ErrorNoSolutions*) non è stata trovata soluzione
- (*CutError.ErrorUnexpected*) uno o entrambi gli indici (*idx*, *idxrct*) non sono validi

3.9.10. NUMBEROFCUTS

La funzione restituisce il numero dei tagli calcolati per un foglio della *soluzione corrente* in corrispondenza dell'indice specificato

int NumberOfCuts(int idx)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione

Valore di ritorno

Il numero di tagli presenti in uno schema di taglio. La funzione restituisce 0 in caso di indice o soluzione non validi.

3.9.11. READRESOLCUT

La funzione restituisce le informazioni relative ad un taglio su un foglio della *soluzione corrente*

int ReadResolCut (int idx, int idxcut, ref double xStart, ref double yStart, ref double xEnd, ref double yEnd, ref int orientation, ref double thickness)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione
- *idxcut*: indice (base zero) del taglio sul foglio della soluzione
- *xStart*: coordinata x del punto d'inizio del taglio
- *yStart*: coordinata y del punto d'inizio del taglio
- *xEnd*: coordinata x del punto di fine del taglio
- *yEnd*: coordinata y del punto di fine del taglio
- *orientation*: orientamento del taglio (0 = verticale, 1 = orizzontale)
- *thickness*: spessore del taglio

Valore di ritorno

Il livello del taglio di cui si sono letti i dati (-1 in caso di errore).

Note

In caso di ritorno -1, è possibile interrogare la proprietà *LastError* per sapere l'errore:

- (*CutError.ErrorKey*) si è verificato esito negativo nella verifica di presenza e stato della chiave
- (*CutError.ErrorNoSolutions*) non è stata trovata soluzione
- (*CutError.ErrorUnexpected*) uno o entrambi gli indici (*idx*, *idxcut*) non sono validi

Il ritorno della funzione indica il *livello di taglio* ed ha valore da 0 a 6.

Le coordinate del taglio sono sempre riferite al sistema cartesiano puro (origine sullo spigolo sinistro-inferiore).

L'argomento *thickness* può assegnare uno spessore differente dallo spessore dell'utensile di taglio:

- *inferiore*, se il taglio è in parte esterno all'area utile di materiale
- *superiore*, se il taglio è da eseguire ripetuto (ad esempio: per aggiunta di un rifilo).

La sequenza dei tagli rispetta la suddivisione della lastra in strisce e le strisce in ulteriori parti: la sequenza rispetta l'ordine secondo cui sono disposti i tagli sul pannello grezzo, a partire dal vertice utilizzato per i piazzamenti verso il vertice opposto.

Non sono assegnati tagli per separare i margini della lastra.

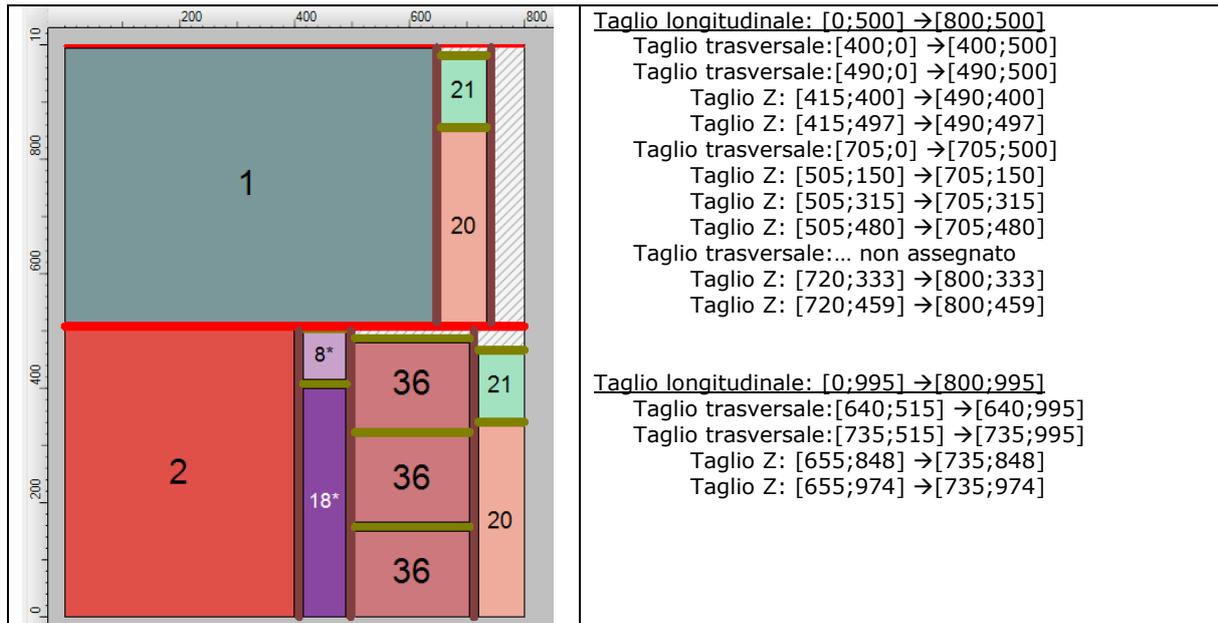
Di seguito un esempio di schema di taglio ottenuto con ottimizzazione in direzione orizzontale:

- il vertice utilizzato per i piazzamenti è *sinistro-inferiore*
- sono presenti 2 tagli longitudinali (tagli di colore: rosso)
- il massimo livello di taglio prodotto è 3 (tagli Z).

La sequenza restituita dei tagli corrisponde alla sequenza riportata a destra dell'immagine.

I righelli posti ai lati del foglio aiutano ad individuare graficamente i tagli.

La lastra grezza ha dimensioni: 800 x 1000 mm.



L'ultimo *taglio trasversale* riportato per il primo *taglio longitudinale* è indicato come "non assegnato", in quanto completamente esterno al margine destro della lastra.

Il *taglio longitudinale* posto sulla parte alta della lastra ha chiaramente uno spessore inferiore al primo, essendo posizionato in sovrapposizione con il bordo superiore della lastra.

3.9.12. CUTLINEAR

La funzione restituisce lo sviluppo lineare corrispondente ai tagli calcolati per un foglio della *soluzione corrente* in corrispondenza dell'indice specificato

double CutLinear(int idx)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione

Valore di ritorno

La funzione restituisce 0 in caso di indice o soluzione non validi; altrimenti: lo sviluppo lineare complessivo dei tagli.

3.9.13. CUTAREA

La funzione restituisce l'area corrispondente ai tagli calcolati per un foglio della *soluzione corrente* in corrispondenza dell'indice specificato

double CutArea(int idx)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione

Valore di ritorno

La funzione restituisce 0 in caso di indice o soluzione non validi, altrimenti: l'area occupata dai tagli.

L'area è calcolata applicando lo spessore dei tagli, come restituito dalla funzione [ReadResolCut](#).

3.9.14. MARGINAREA

La funzione restituisce l'area corrispondente ai margini assegnati per un foglio della *soluzione corrente* in corrispondenza dell'indice specificato

double MarginArea(int idx)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione

Valore di ritorno

La funzione restituisce 0 in caso di indice o soluzione non validi oppure di margini assegnati nulli; altrimenti: l'area corrispondente ai margini.

I margini sono assegnati in struttura [OneSheet](#) del foglio.

3.9.15. GETWASTE

La funzione restituisce l'area non utilizzata per un foglio della *soluzione corrente* in corrispondenza dell'indice specificato

double GetWaste (int idx)

Argomenti

- *idx*: indice (base zero) del foglio della soluzione

Valore di ritorno

La funzione restituisce 0 in caso di indice o soluzione non validi, altrimenti: l'area non utilizzata del foglio.

L'area non utilizzata di un foglio è calcolata sottraendo all'area originale del foglio:

- l'area corrispondente ai margini
- l'area corrispondente ai rettangoli tagliati
- l'area dei tagli.

3.9.16. ESTIMATEDTIME

Assegna il tempo stimato dal client per eseguire lo schema di taglio della *soluzione corrente* in corrispondenza dell'indice specificato

public bool EstimatedTime(int idx, long valueH, long valueM, long valueS)

Argomenti

- *idx*: indice della lastra. Se -1: assegna il tempo totale della soluzione.
- *valueH*: numero di ore (≥ 0)
- *valueM*: numero di minuti (≥ 0)
- *valueS*: numero di secondi (≥ 0)

Valore di ritorno

true se l'esito è positivo.

Note

Un ritorno differente da *true* corrisponde ad una delle situazioni di errore:

- nessuna soluzione calcolata
- indice (*idx*) non valido
- valori negativi per i valori temporali.

L'assegnazione del tempo in (ore, minuti, secondi) permette una maggiore flessibilità: è comunque possibile assegnare, ad esempio, il tempo complessivo in secondi.

La funzione permette di assegnare una informazione che è determinabile solo sulla base di valutazioni che dipendono dall'applicazione specifica.

3.9.17. ESTIMATEDCOST

Assegna il costo stimato dal client per eseguire lo schema di taglio della *soluzione corrente* in corrispondenza dell'indice specificato

public bool EstimatedCost(int idx, double value)

Argomenti

- *idx*: indice della lastra. Se -1: assegna il costo totale della soluzione
- *value*: costo (della lastra o soluzione)

Valore di ritorno

true se l'esito è positivo.

Note

Un ritorno differente da *true* corrisponde ad una delle situazioni di errore:

- nessuna soluzione calcolata
- indice (*idx*) non valido
- valore negativo per il costo

La funzione permette di assegnare una informazione che è determinabile solo sulla base di valutazioni che dipendono dall'applicazione specifica.

3.9.18. EXPORT

La funzione salva la *soluzione corrente* in uno o più files (formato XML).

CutError Export (string pathName)

Argomenti

- *pathName*: percorso del file

Valore di ritorno

CutError.ErrorNone se l'esito è positivo.

Note

Un ritorno differente da *CutError.ErrorNone* corrisponde ad una delle situazioni di errore:

- (*CutError.ErrorNoSolutions*) non risulta in presa una soluzione
- (*CutError.ErrorKey*) licenza non valida o chiave non trovata
- (*CutError.ErrorIOfile*) si è verificato un errore nell'accesso al file da scrivere
- (*CutError.ErrorUnexpected*) si è verificato un errore inaspettato.

I files salvati possono essere interpretati da un applicativo esterno in alternativa alla interrogazione delle funzioni esposte. In particolare:

- I files hanno formato "Xml": l'eventuale estensione ".XML" è aggiunta a *pathName*, se necessario
- è salvato un file per ogni schema di taglio calcolato
- ogni file contiene l'informazione dell'eventuale ripetizione dello schema di taglio
- il primo file è salvato con nome *pathName+ "_001"*
 - il secondo file è salvato con nome *pathName+ "_002"*
 - fino ad esaurimento degli schemi di taglio
 - files esistenti sono sovrascritti.

STRUTTURA DI FILE CORRISPONDENTE AD UNO SCHEMA DI TAGLIO

Il nodo principale è il nodo **MAIN** e contiene le informazioni di carattere generale distribuite in elementi innestati:

```
<MAIN>
<GENERALSETTINGS Info1="Client ABC" Info2="ABC12345" />
<TECHSETTINGS Unit="0" Direction="0" Corner="0" MaxCutLevel="3" />
<DIM Code="3.0" L="800.0" H="1000.0" T="0.0" G="N" Scrap="0" Priority="0" />
<DIMTRIMS BladeThickness="15.0" PreCut="0.0" LongCut="0.0" TransvCut="0.0" ZCut="0.0" />
<DATA Rep="2" />
<PIECESLIST>
<PIECE Code="1" L="640.0" H="480.0" T="0.0" Used="1" Rotation="1" G="N" Priority="0" .../>
<PIECE Code="2" L="400.0" H="500.0" T="0.0" Used="1" Rotation="1" G="N" Priority="1" ... />
...
</PIECESLIST>
```

```
<DRAW>
...
</DRAW>
</MAIN>
```

NODO <GENERALSETTINGS>

riporta le informazioni generiche del progetto (vedi: [Custom1, Custom2, ..., Custom10](#))

- attributi: *Info1, ..., Info10*

NODO <TECHSETTINGS>

riporta alcune assegnazioni del progetto

- Attributo *Unit*: unità di misura del progetto (vedi: [Unit](#))
- Attributo *Direction*: direzione di avanzamento per i piazzamenti (vedi: [Direction](#))
- Attributo *Corner*: vertice di partenza per i piazzamenti (vedi: [Corner](#))
- Attributo *MaxCutLevel*: il massimo livello di taglio utilizzabile in ottimizzazione (vedi: [MaxCutLevels](#))

NODO <DIM>

riporta le informazioni generali della lastra (vedi struttura: [OneSheet](#))

- Attributo *Code*: campo *ID* di struttura
- Attributo *L*: campo *Length* di struttura
- Attributo *H*: campo *Height* di struttura
- Attributo *T*: campo *Thickness* di struttura
- Attributo *Descr*: campo *Label* di struttura
- Attributo *Mat*: campo *Material* di struttura
- Attributo *G*: campo *Grain* di struttura (valori: "N"=nessuna venatura, "X"=venatura x, "Y"=venatura y)
- Attributo *Scrap*: campo *Scrap* di struttura
- Attributo *Priority*: campo *Priority* di struttura
- Attributo *TopMargin*: campo *BorderTop* di struttura
- Attributo *BottomMargin*: campo *BorderBottom* di struttura
- Attributo *RightMargin*: campo *BorderRight* di struttura
- Attributo *LeftMargin*: campo *BorderLeft* di struttura
- Attributo *Cost*: campo *Cost* di struttura

NODO <DIMTRIMS>

riporta le assegnazioni dei rifili del progetto

- Attributo *BladeThickness*: spessore dell'utensile di taglio (vedi: [CutterDiameter](#))
- Attributo *PreCut*: rifilo aggiunto a tutti i pre-tagli (vedi: [PreCut](#))
- Attributo *LongCut*: rifilo aggiunto ai tagli longitudinali(vedi: [Longcut](#))
- Attributo *TransvCut*: rifilo aggiunto ai tagli trasversali (vedi: [Transvcut](#))
- Attributo *ZCut*: rifilo aggiunto ai tagli Z (vedi: [Zcut](#))
- Attributo *LongCutTFC*: distanza aggiuntiva tra i tagli di primo livello allo scopo di alleviare la tensione della lastra (vedi: [TensionGap](#))

NODO <DATA>

riporta le informazioni generali relative allo schema di taglio

- Attributo *Rep*: ripetizione del foglio
- Attributo *Cost*: costo stimato di esecuzione (come assegnato dal client)
- Attributo *Time*: tempo stimato di esecuzione (unità: secondi)- (come assegnato dal client)

NODO <PIECESLIST>

- riporta l'elenco dei rettangoli che risultano tagliati dallo schema di taglio corrente. Ad ognuno corrisponde un nodo <PIECE>, con riportate le informazioni generali del rettangolo (vedi struttura: [OneRect](#)):
 - Attributo *Code*: campo *ID* di struttura
 - Attributo *L*: campo *Length* di struttura
 - Attributo *H*: campo *Height* di struttura
 - Attributo *T*: campo *Thickness* di struttura
 - Attributo *Descr*: campo *Label* di struttura
 - Attributo *Mat*: campo *Material* di struttura
 - Attributo *G*: campo *Grain* di struttura (valori: "N"=nessuna venatura, "X"=venatura x, "Y"=venatura y)
 - Attributo *Rotation*: campo *Rotate* di struttura
 - Attributo *Priority*: campo *Priority* di struttura
 - Attributi *MatEdge1*, *ThickEdge1*, *EnableEdge1*: campi *MatEdge1*, *ThickEdge1*, *BoolEdge1* di struttura
 - Attributi *MatEdge2*, *ThickEdge2*, *EnableEdge2*: campi *MatEdge2*, *ThickEdge2*, *BoolEdge2* di struttura
 - Attributi *MatEdge3*, *ThickEdge3*, *EnableEdge3*: campi *MatEdge3*, *ThickEdge3*, *BoolEdge3* di struttura
 - Attributi *MatEdge4*, *ThickEdge4*, *EnableEdge4*: campi *MatEdge4*, *ThickEdge4*, *BoolEdge4* di struttura
 - Attributi *Param1*, ..., *Param15*: campi (*Param1*, ..., *Param15*) di struttura
- oltre a:
- Attributo *Used*: numero di piazzamenti nel foglio corrente

NODO <DRAW>

riporta lo schema di taglio corrispondente alla lastra, ma con modalità differenti da quanto visto in precedenza.

Di seguito lo schema di taglio corrispondente all'esempio riportato in paragrafo [ReadResolCut](#)

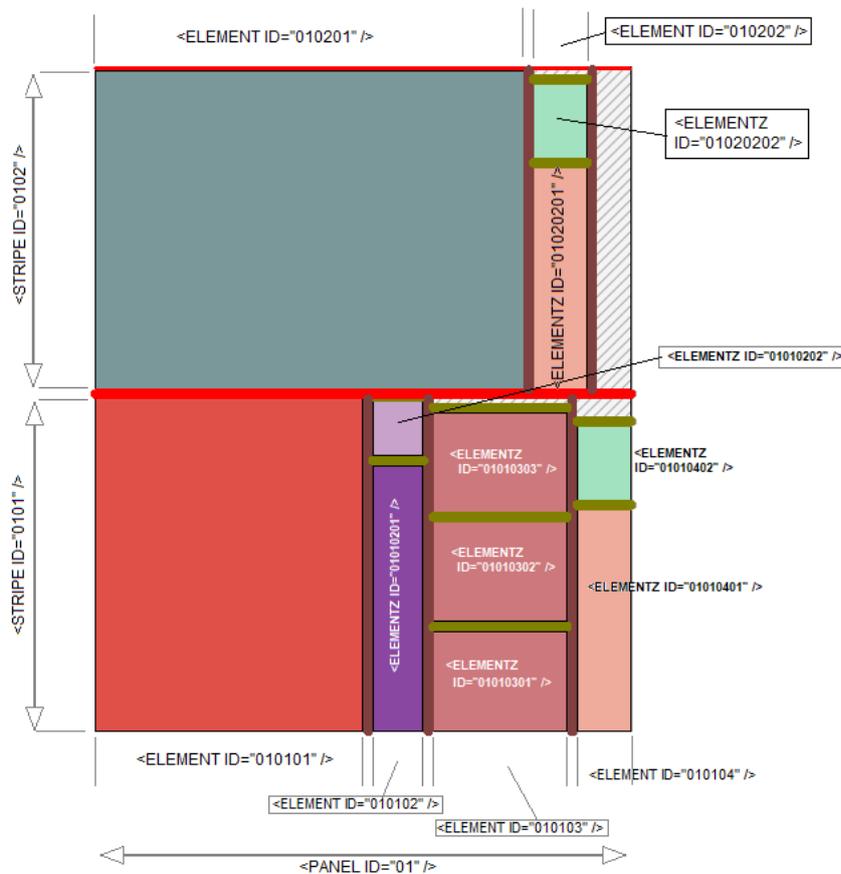
```
<DRAW>
<PANEL ID="01" REP="1" L="800.00" H="1000.000">
  <STRIPE ID="0101" REP="1" L="800.000" H="500.000">
    <ELEMENT ID="010101" REP="1" L="400.000" H="500.000">
      <LABEL Code="2" Rep="1" Rotated="0" />
    </ELEMENT>
    < ELEMENT ID="010102" REP="1" L="75.000" H="500.000">
      < ELEMENTZ ID="01010201" REP="1" L="75.000" H="400.000">
        <LABEL Code="18" Rep="1" Rotated="1" />
      </ELEMENTZ>
      < ELEMENTZ ID="01010202" REP="1" L="75.000" H="82.000">
        <LABEL Code="8" Rep="1" Rotated="1" />
      </ELEMENTZ>
    </ELEMENT>
    < ELEMENT ID="010103" REP="1" L="200.000" H="500.000">
      <ELEMENTZ ID="01010301" REP="1" L="200.000" H="150.000">
        <LABEL Code="36" Rep="1" Rotated="0" />
      </ELEMENTZ>
      <ELEMENTZ ID="01010302" REP="1" L="200.000" H="150.000">
        <LABEL Code="36" Rep="1" Rotated="0" />
      </ELEMENTZ>
      <ELEMENTZ ID="01010303" REP="1" L="200.000" H="150.000">
        <LABEL Code="36" Rep="1" Rotated="0" />
      </ELEMENTZ>
    </ELEMENT>
    <ELEMENT ID="010104" REP="1" L="80.000" H="500.000">
      <ELEMENTZ ID="01010401" REP="1" L="80.000" H="333.000">
        <LABEL Code="20" Rep="1" Rotated="0" />
      </ELEMENTZ>
      <ELEMENTZ ID="01010402" REP="1" L="80.000" H="111.000">
        <LABEL Code="21" Rep="1" Rotated="0" />
      </ELEMENTZ>
    </ELEMENT>
  </STRIPE>
  <STRIPE ID="0102" REP="1" L="800.000" H="480.000">
    <ELEMENT ID="010201" REP="1" L="640.000" H="480.000">
```

```

<LABEL Code="1" Rep="1" Rotated="0" />
</ELEMENT>
<ELEMENT ID="010202" REP="1" L="80.000" H="480.000">
  <ELEMENTZ ID="01020201" REP="1" L="80.000" H="333.000">
    <LABEL Code="20" Rep="1" Rotated="0" />
  </ELEMENTZ>
  <ELEMENTZ ID="01020202" REP="1" L="80.000" H="111.000">
    <LABEL Code="21" Rep="1" Rotated="0" />
  </ELEMENTZ>
</ELEMENT>
</STRIPE>
</PANEL>
</DRAW>

```

La figura evidenzia il significato attribuito ad ogni singolo nodo riportato nello schema "xml":



NODO <PANEL>

Dato il pannello grezzo, questo può essere separato con pretagli (tagli in direzione verticale, a copertura dell'altezza del pannello grezzo) in più pannelli: come già detto, è il caso di pannello ottimizzato applicando *direzione verticale*.

Ogni elemento così ottenuto corrisponde ad una sezione di tipo **<PANEL>**: in caso di pannello ottimizzato applicando *direzione orizzontale*, la sezione è sempre una (come in esempio).

L'ordine delle sezioni di tipo **<PANEL>** rispetta l'ordine secondo cui sono disposte sul pannello grezzo, a partire dal vertice utilizzato per i piazzamenti verso il vertice opposto (nell'esempio: dal vertice *sinistro-inferiore* verso il vertice *destra-superiore*).

Gli attributi del nodo sono comuni a tutte le tipologie di nodi successivi, sia nel nome che nel significato.

```
<PANEL ID="01" REP="1" L="800.00" H="1000.000">
```

- Attributo **ID**: identificativo univoco
- Attributo **REP**: ripetizione dell'elemento (default= 1)

- Attributo *L*: dimensione orizzontale dell'elemento (al netto dei tagli)
- Attributo *H*: dimensione verticale dell'elemento (al netto dei tagli)

Nel nostro esempio: l'elemento ha le stesse dimensioni del pannello grezzo (800 x 1000 mm).

NODO <STRIPE>

Nodi inseriti all'interno di un nodo <PANEL>, a descrizione delle *strisce* generate dai tagli longitudinali (tagli in direzione orizzontale, a copertura della lunghezza dell'elemento di appartenenza <PANEL>). L'ordine delle sezioni di tipo <STRIPE> rispetta l'ordine secondo cui sono disposte sull'elemento <PANEL>, nel rispetto della direzione imposta dal vertice utilizzato per i piazzamenti (nell'esempio: dal basso verso l'alto).

Gli attributi del nodo sono gli stessi esaminati per il nodo padre.

Nell'esempio risultano assegnate 2 sezioni <STRIPE>.

```
<DRAW>
<PANEL ID="01" REP="1" L="800.00" H="1000.000">
  <STRIPE ID="0101" REP="1" L="800.000" H="500.000">
  ...
  </STRIPE>
  <STRIPE ID="0102" REP="1" L="800.000" H="480.000">
  ...
  </STRIPE>
</PANEL>
</DRAW>
```

NODO <ELEMENT>

Nodi inseriti all'interno di un nodo <STRIPE>, a descrizione degli elementi generati dai tagli trasversali (tagli in direzione verticale, a copertura dell'altezza dell'elemento di appartenenza <STRIPE>). L'ordine delle sezioni di tipo <ELEMENT> rispetta l'ordine secondo cui sono disposte sull'elemento <STRIPE>, nel rispetto della direzione imposta dal vertice utilizzato per i piazzamenti (nell'esempio: da sinistra verso destra).

Gli attributi del nodo sono gli stessi esaminati per il nodo padre.

Nell'esempio risultano assegnate:

- 4 sezioni per la prima striscia
- 2 sezioni per la seconda striscia.

```
<PANEL ID="01" REP="1" L="800.00" H="1000.000">
  <STRIPE ID="0101" REP="1" L="800.000" H="500.000">
    <ELEMENT ID="010101" REP="1" L="400.000" H="500.000">
      <LABEL Code="2" Rep="1" Rotated="0" />
    </ELEMENT>
    <ELEMENT ID="010102" REP="1" L="75.000" H="500.000">
    ...
  </ELEMENT>
  <ELEMENT ID="010103" REP="1" L="200.000" H="500.000">
  ...
  </ELEMENT>
  <ELEMENT ID="010104" REP="1" L="80.000" H="500.000">
  ...
  </ELEMENT>
  </STRIPE>
  <STRIPE ID="0102" REP="1" L="800.000" H="480.000">
    <ELEMENT ID="010201" REP="1" L="640.000" H="480.000">
    ...
  </ELEMENT>
  <ELEMENT ID="010202" REP="1" L="80.000" H="480.000">
  ...
  </ELEMENT>
</STRIPE>
</PANEL>
```

NODO <ELEMENTZ>

Nodi inseriti all'interno di un nodo <ELEMENT>, a descrizione degli elementi generati da tagli Z (tagli in direzione orizzontale, a copertura della lunghezza dell'elemento di appartenenza <ELEMENT>). L'ordine delle sezioni di tipo <ELEMENTZ> rispetta l'ordine secondo cui sono disposte sull'elemento <ELEMENT>, nel rispetto della direzione imposta dal vertice utilizzato per i piazzamenti (nell'esempio: dal basso verso l'alto).

NODO <ELEMENTW>

Nodi inseriti all'interno di un nodo <ELEMENTZ>, a descrizione degli elementi generati da tagli W (tagli in direzione verticale, a copertura dell'altezza dell'elemento di appartenenza <ELEMENTZ>). L'ordine delle sezioni di tipo <ELEMENTW> rispetta l'ordine secondo cui sono disposte sull'elemento <ELEMENTZ>, nel rispetto della direzione imposta dal vertice utilizzato per i piazzamenti (nell'esempio: da sinistra verso destra).

Nell'esempio non risultano assegnati nodi di questo tipo.

NODO <ELEMENT5>

Nodi inseriti all'interno di un nodo <ELEMENTW>, a descrizione degli elementi generati da tagli #5 (tagli in direzione orizzontale, a copertura della lunghezza dell'elemento di appartenenza <ELEMENTW>). L'ordine delle sezioni di tipo <ELEMENT5> rispetta l'ordine secondo cui sono disposte sull'elemento <ELEMENTW>, nel rispetto della direzione imposta dal vertice utilizzato per i piazzamenti (nell'esempio: dal basso verso l'alto).

Nell'esempio non risultano assegnati nodi di questo tipo.

NODO <ELEMENT6>

Nodi inseriti all'interno di un nodo <ELEMENT5>, a descrizione degli elementi generati da tagli #6 (tagli in direzione verticale, a copertura dell'altezza dell'elemento di appartenenza <ELEMENT5>). L'ordine delle sezioni di tipo <ELEMENT6> rispetta l'ordine secondo cui sono disposte sull'elemento <ELEMENT5>, nel rispetto della direzione imposta dal vertice utilizzato per i piazzamenti (nell'esempio: da sinistra verso destra).

Nell'esempio non risultano assegnati nodi di questo tipo.

NODO <LABEL>

Un nodo di questo tipo può essere inserito all'interno di ogni nodo di tipo (PANEL, STRIPE, ELEMENT, ELEMENTZ, ELEMENTW, ELEMENT5, ELEMENT6), a definizione del rettangolo che risulta piazzato nell'elemento corrispondente. Un nodo <LABEL> termina un ramo di taglio.

```
<PANEL ID="01" REP="1" L="800.00" H="1000.000">
  <STRIPE ID="0101" REP="1" L="800.000" H="500.000">
    <ELEMENT ID="010101" REP="1" L="400.000" H="500.000">
      <LABEL Code="2" Rep="1" Rotated="0" />
    </ELEMENT>
  ...
</STRIPE>
<STRIPE ID="0102" REP="1" L="800.000" H="480.000">
  ...
</STRIPE>
</PANEL>
```

Il nodo assegna i seguenti attributi:

- Attributo *Code*: identificativo del rettangolo (campo ID in struttura [OneRect](#))
- Attributo *REP*: ripetizione dell'elemento (default= 1)
- Attributo *Rotated*: indica se il rettangolo è piazzato ruotato.

3.10. FUNZIONI DI SERIALIZZAZIONE DEL PROGETTO

Le funzioni gestiscono la serializzazione del progetto su/da file.

3.10.1. SAVEPROJECT

La funzione salva le assegnazioni delle liste di *rettangoli* e *fogli* in file (formato XML).

CutError SaveProject (string pathName, bool bMode)

Argomenti

- *pathName*: percorso del file
- *bMode*: *true* richiede il salvataggio dei settaggi generali di progetto

Valore di ritorno

CutError.ErrorNone se l'esito è positivo.

Note

Un ritorno differente da *CutError.ErrorNone* corrisponde ad una delle situazioni di errore:

- (*CutError.ErrorIOFile*) si è verificato un errore nell'accesso al file da scrivere.

Utilizzare *bMode=true* per salvare anche i settaggi generali di progetto.

3.10.2. LOADPROJECT

La funzione legge le assegnazioni delle liste di *rettangoli* e *fogli* da file (formato XML).

CutError LoadProject (string pathName, bool bMode)

Argomenti

- *pathName*: percorso del file
- *bMode*: *true* abilita la lettura dei settaggi generali di progetto

Valore di ritorno

CutError.ErrorNone se l'esito è positivo.

Note

Un ritorno differente da *CutError.ErrorNone* corrisponde ad una delle situazioni di errore:

- (*CutError.ErrorIOFile*) Errore nell'accesso del file.

Utilizzare *bMode=true* per leggere anche i settaggi generali di progetto: i settaggi che non sono letti dal file sono inizializzati ai valori di default.

Se *bMode=false*: i settaggi rimangono invariati.

L'esecuzione della funziona non modifica i settaggi generali di ottimizzazione.

3.10.3. IMPORTPROJECT

La funzione importa le assegnazioni delle liste di *fogli* e *rettangoli* da file CSV

CutError ImportProject (string pathName, string stFormatSheet, string stFormatRect)

Argomenti

- *pathName*: percorso del file
- *stFormatSheet*: stringa di formattazione della riga di assegnazione foglio (in file CSV)
- *stFormatRect*: stringa di formattazione della riga di assegnazione rettangolo (in file CSV)

Valore di ritorno

CutError.ErrorNone se l'esito è positivo.

Note

Un ritorno differente da *CutError.ErrorNone* corrisponde ad una delle situazioni di errore:

- (*CutError.ErrorIOFile*) Errore nell'accesso del file
- (*CutError.ErrorFileNotValid*) dati letti inconsistenti

La funziona inizializza il progetto attuale assegnando le liste di *fogli e rettangoli* da file CSV e lasciando inalterati i restanti settaggi generali di progetto.

Un file CSV è un file di testo utilizzato per l'assegnazione di una tabella di dati. Ogni riga del file corrisponde ad una riga di tabella ed è a sua volta suddivisa in campi (singole colonne) attraverso un carattere separatore (sono riconosciuti validi: la virgola, il punto e virgola).

La funzione assegna due stringhe a formattazione delle righe di assegnazione di ogni tipo di elemento, che sono utilizzate se lo stesso file CSV non ne assegna altre.

Vediamo un esempio di file CSV:

<pre>***panel;d;l;h;n;m;g;p ***part;d;l;h;n;x;m;g;p;r</pre>	<p>linee di formattazione elementi</p> <ul style="list-style-type: none"> • opzionali • non necessariamente presenti entrambe • l'intestazione riconosciuta per le linee è evidenziata: <ul style="list-style-type: none"> ○ ***panel; per i fogli ○ ***part; per i rettangoli
<pre>panel;MDF;2000;1000;1; panel;MDF2;2000;1500;1;</pre>	<p>linee di assegnazione fogli</p> <ul style="list-style-type: none"> - l'intestazione riconosciuta per le linee è evidenziata: panel; - i campi successivi sono interpretati sulla base della formattazione qui assegnata nella prima linea del file
<pre>part;A1;1000;300;30;0;;;1 part;A2;750;250;30;0;;;1 part;A3;300;250;30;0;;;1 part;A4;150;130;50;10;;;1 part;A5;180;150;0;20;;;0</pre>	<p>Linee di assegnazione rettangoli</p> <ul style="list-style-type: none"> - l'intestazione riconosciuta per le linee è evidenziata: part; - i campi successivi sono interpretati sulla base della formattazione qui assegnata nella seconda linea del file

Una stringa di formattazione è ritenuta valida se assegna i campi corrispondenti a: lunghezza, altezza, quantità. Ogni campo è contrassegnato con una stringa univoca di lunghezza massima di due caratteri. Vediamo la lista completa:

- **e** abilitazione di riga (interpreta: 0/1, yes/no, on/off, true/false) – (default=1)
- **d** descrizione dell'elemento – (default="")
- **l** dimensione di lunghezza dell'elemento
- **h** dimensione di altezza dell'elemento
- **s** dimensione di spessore dell'elemento
- **n** quantità dell'elemento
- **x** quantità extra (solo per rettangoli)
- **m** materiale
- **g** venatura (interpreta: 1/2, x/y,) – (default=0)
- **r** possibilità di rotazione (solo per rettangoli) (interpreta: 0/1, yes/no, on/off, true/false) – (default=0)
- **p** priorità
- **et** bordo superiore di rettangolo (formato massimo: "spessore|nome|flag di taglio". ES: "3.0|abc456|0")
- **eb** bordo inferiore di rettangolo (formato: vedi campo precedente)
- **er** bordo destro di rettangolo (formato: vedi campo precedente)
- **el** bordo sinistro di rettangolo (formato: vedi campo precedente)
- **a1** campo *Param1* di rettangolo
- **a2;a3;a4;a5;a6;a7;a8;a9;a10;a11;a12;a13;a14;a15** campi (*Param2,...,Param15*) di rettangolo
- **mt** margine superiore di lastra
- **mb** margine inferiore di lastra
- **mr** margine destro di lastra
- **ml** margine sinistro di lastra
- **y** informazione di lastra di recupero (interpreta: 0/1, yes/no, on/off, true/false).

Nell'esempio sopra riportato:

- la stringa di formattazione delle lastre assegna "d;l;h;n;m;g;p":
 - "d": assegna il campo *Label* in struttura [OneSheet](#)
 - "l": assegna il campo *Length* in struttura
 - "h": assegna il campo *Height* in struttura
 - "n": assegna il campo *N* in struttura
 - "m": assegna il campo *Material* in struttura
 - "g": assegna il campo *Grain* in struttura
 - "p": assegna il campo *Priority* in struttura
- la stringa di formattazione dei rettangoli assegna "d;l;h;n;x;m;g;p;r":
 - "d": assegna il campo *Label* in struttura [OneRect](#)
 - "l": assegna il campo *Length* in struttura
 - "h": assegna il campo *Height* in struttura
 - "n": assegna il campo *N* in struttura
 - "x": assegna il campo *Extra* in struttura
 - "m": assegna il campo *Material* in struttura
 - "g": assegna il campo *Grain* in struttura
 - "p": assegna il campo *Priority* in struttura
 - "r": assegna il campo *Rotate* in struttura.

4. GUIDA ALL'UTILIZZO DELLA LIBRERIA

Scopo del capitolo è fornire le informazioni necessarie per l'integrazione della libreria TPA_C nella vostra applicazione.

4.1. DIAGRAMMA DI FLUSSO TIPICO

La sequenza riportata di seguito descrive un utilizzo di base della libreria:

1. creare una istanza della classe *TpaCutOEM.TpaRctCut()*
2. effettuare le verifiche preliminari di funzionamento generale (verifica della presenza della chiave)
3. assegnazione dei settaggi di funzionamento generale (vedi: [NumberCustom](#), [NumberRectParam](#), ...)
4. assegnazione dei settaggi generali di progetto (vedi: [Unit](#), [Direction](#), ...)
5. assegnazione dei *rettangoli* con chiamate della funzione [AddRct\(\)](#)
6. assegnazione dei *fogli* con chiamate della funzione [AddSheet\(\)](#)
7. avvio della procedura di ottimizzazione con chiamata della funzione [Compute \(\)](#)
8. acquisizione dei risultati con chiamata alle funzioni: [NumberOfSolutions](#), [SelectSolution](#), ...
9. acquisizioni/elaborazioni autonome delle informazioni di efficienza dell'ottimizzazione ed assegnazioni alla classe ([EstimatedTime](#), [EstimatedCost](#))
10. salvataggio report di ottimizzazione ([Export](#)).

L'ordine di assegnazione di *settaggi*, *rettangoli* e *fogli* è indifferente: quello proposto è solo indicativo.

L'insieme di settaggi generali, rettangoli e fogli costituiscono quello che è indicato come *progetto di taglio*, in quanto identifica l'insieme dei dati necessari all'esecuzione di una ottimizzazione.

4.2. VERIFICHE PRELIMINARI

Una verifica preliminare della presenza della chiave può essere utile e necessario per adattare le funzionalità del vostro applicativo:

- verificare la presenza e validità della chiave con interrogazione di [IsValidLicense](#)

Si rammenta che la verifica della chiave è eseguita con scadenza sia temporale (cioè: ogni X secondi) che contestuale (cioè: con chiamata a funzioni specifiche): al fine di garantire un normale funzionamento si raccomanda quindi di non estrarre la chiave durante l'esecuzione del vostro applicativo.

4.3. ASSEGNAZIONE DEI SETTAGGI DI FUNZIONAMENTO GENERALE

È generalmente necessario effettuare una assegnazione preliminare dei settaggi di funzionamento generale. Si tratta di informazioni che non cambiano al variare del progetto, trattandosi di impostazioni generalmente attribuibili ad opzioni di funzionamento generale di una applicazione.

4.4. ASSEGNAZIONE DEI SETTAGGI GENERALI DI PROGETTO

È necessario effettuare una assegnazione preliminare dei settaggi generali di progetto, che possono cambiare per ogni progetto.
L'assegnazione preliminare dei settaggi può anche avvenire con lettura da un file salvato in precedenza: vedi funzioni [SaveProject\(\)](#), [LoadProject\(\)](#).

4.5. ASSEGNAZIONE DEI RETTANGOLI

Si passa all'assegnazione della lista di rettangoli.

È opportuno sottolineare alcuni punti fondamentali nell'assegnazione dei rettangoli:

- ogni rettangolo ha un identificativo numerico univoco, strettamente positivo (> 0): campo *ID* in struttura *OneRect*
- non è quindi possibile assegnare lo stesso *ID* per più rettangoli
- gli identificativi possono essere assegnati in ordine qualunque e non necessariamente consecutivi
- occorre tenere presente che gli *ID* hanno un ruolo importante per l'ordinamento dei rettangoli: a parità di altre impostazioni notevoli (Es.: priorità, area del rettangolo) è l'*ID* che decide quale rettangolo ha una maggiore priorità di piazzamento nel processo di ottimizzazione.

Per aggiungere un rettangolo chiamare la funzione [AddRct\(\)](#).

Per eliminare i rettangoli chiamare la funzione [ClearRct\(\)](#).

4.6. ASSEGNAZIONE DEI FOGLI

Si passa all'assegnazione della lista dei fogli.

Come già per i rettangoli, è opportuno sottolineare alcuni punti fondamentali nell'assegnazione dei fogli:

- ogni foglio ha un identificativo numerico univoco, strettamente positivo (> 0): campo *ID* in struttura *OneSheet*
- non è quindi possibile assegnare lo stesso *ID* per più fogli
- gli identificativi possono essere assegnati in ordine qualunque e non necessariamente consecutivi
- occorre tenere presente che gli *ID* hanno un ruolo importante per l'ordinamento dei fogli: a parità di altre impostazioni notevoli (Es.: priorità, informazione di sfrido, dimensioni) è l'*ID* che decide quale foglio ha una maggiore priorità di utilizzo nel processo di ottimizzazione.

Per aggiungere un foglio chiamare la funzione [AddSheet\(\)](#).

Per eliminare la lista dei fogli chiamare la funzione [ClearSheet\(\)](#).

4.7. ESECUZIONE DELL'OTTIMIZZAZIONE DI TAGLIO

Per avviare una ottimizzazione del progetto è necessario chiamare la funzione [Compute](#).

Prima dell'ottimizzazione sono eseguite verifiche sui rettangoli e sui fogli che possono anche determinare un ritorno della funzione con errore e conseguente annullamento dell'ottimizzazione:

- devono risultare rettangoli richiesti per i piazzamenti e fogli disponibili
- la verifica dei filtri di corrispondenza (spessore, materiale) deve potere abbinare almeno un rettangolo ad un foglio.

4.7.1. ACQUISIRE IL RISULTATO DELLA SOLUZIONE

Ad ottimizzazione eseguita è possibile acquisire i risultati della soluzione di taglio.

Un gruppo di funzioni permette invece di acquisire tutte le informazioni relative alla soluzione dell'ottimizzazione:

- *NumberOfSolutions*: numero di soluzioni calcolate dall'ottimizzatore
- *FitnessSheet*: efficienza della soluzione
- *SolutionSheets*: informazione del numero di schemi di taglio della soluzione
- *UsedRects*: informazione del numero di piazzamenti della soluzione o di un foglio e/o di un tipo di rettangolo
- *ReadResolRect*: informazioni sui rettangoli relative a un singolo foglio della soluzione
- *ReadResolCut*: informazioni sui tagli relative a un singolo foglio della soluzione
- *Export*: salva la soluzione su file (formato XML)

4.7.1.1. CODICE DI ESEMPIO: COME ACQUISIRE LE INFORMAZIONI GENERALI DEI FOGLI DELLA SOLUZIONE

```

TpaCutOEM.TpaRctCut optimizeObj=new TpaCutOEM.TpaRctCut ();
...
//Campi di assegnazione su interrogazione del componente
OneRect OneRect = new OneRect();

//ciclo di interrogazioni sui fogli della soluzione
int IndexSheet = 0; //indice di foglio
for (IndexSheet = 0; IndexSheet < optimizeObj.SolutionSheets; IndexSheet++)
{
    //ID del foglio di index (IndexSheet)
    int IdSheet = optimizeObj.GetSheetID(IndexSheet);
    //ripetizione del foglio
    int sheetRepetition = optimizeObj.NumberOfRepetitions(IndexSheet);
    // ...
    //ciclo di acquisizione dei piazzamenti di un foglio
    //.....
}

```

4.7.1.2. CODICE DI ESEMPIO: CICLO DI ACQUISIZIONE DEI PIAZZAMENTI DI UN FOGLIO

```

// IndexSheet = indice di foglio (vedi: ciclo precedente)
int IndexRct= 0; //indice di piazzamento sul foglio
OneRect oneRect=new OneRect();

for (IndexRct = 0; IndexRct < UsedRects(IndexSheet, 0); IndexRct ++ )
{
    double qx = 0.0, qy = 0.0;
    bool rotate = false;
    //legge posizione e rotazione del pezzo inserito
    int Id = optimizeObj.ReadResolRect(IndexSheet, IndexRct, ref qx, ref qy, ref rotate);
    //legge le dimensioni e caratteristiche del pezzo inserito
    optimizeObj.ReadRect(Id, ref OneRect);
    //...
}

```

4.7.1.3. CODICE DI ESEMPIO: CICLO DI ACQUISIZIONE DEI TAGLI DI UN FOGLIO

```
// IndexSheet = indice di foglio (vedi: ciclo precedente)
for (idxP = 0; idxP < optimizeObj.NumberOfCuts(IndexSheet); idxP++)
{
    double xStart = 0, yStart = 0, xEnd = 0, yEnd = 0, thickness = 0;
    int orientation = 0;
    int levelCut = optimizeObj.ReadResolCut(i, idxP, ref xStart, ref yStart, ref xEnd, ref yEnd,
        ref orientation, ref thickness);
    //...
}
```

4.7.2. GESTIRE SOLUZIONI MULTIPLE

È possibile che l'ottimizzatore trovi più soluzioni, fino ad un massimo di 5. Ad ottimizzazione eseguita, la prima soluzione tra quelle calcolate è impostata in automatico come *soluzione attuale*.

Tutte le soluzioni calcolate rimangono disponibili, con possibilità di navigare tra le medesime cambiando la soluzione corrente su opportuna specifica.

La soluzione corrente può essere salvata su file chiamando la funzione [Export](#).

4.7.2.1. CODICE DI ESEMPIO: NAVIGAZIONE TRA SOLUZIONI MULTIPLE

```
TpaCutOEM.TpaRctCut optimizeObj=new TpaCutOEM.TpaRctCut();

// GoToNextSolution: passa alla soluzione successiva calcolata
bool GoToNextSolution()
{
    If (optimizeObj.SelectSolution < optimizeObj.NumberOfSolution)
    {
        optimizeObj.SelectSolution = optimizeObj.SelectSolution+1;
        return true;
    }
    return false;
}

// GoToPrevSolution: passa alla soluzione precedente calcolata
bool GoToPrevSolution()
{
    If (optimizeObj.SelectSolution >0)
    {
        optimizeObj.SelectSolution = optimizeObj.SelectSolution-1;
        return true;
    }
    return false;
}
```

4.8. SALVARE E LEGGERE UN PROGETTO DI TPA_C

È possibile salvare e recuperare un progetto mediante le funzioni [SaveProject](#) e [LoadProject](#).

Le due funzioni permettono di salvare e recuperare anche i settaggi generali di un progetto.

Dopo l'esecuzione di *LoadProject*, un applicativo esterno deve aggiornare settaggi di progetto, rettangoli e fogli, con lettura delle informazioni da TPA_C.

4.8.1. CODICE DI ESEMPIO



```
TpaCutOEM.TpaRctCut optimizeObj=new TpaCutOEM.TpaRctCut ();

//Strutture
OneRect ItemRect=new OneRect();
OneSheet ItemSheet=new OneSheet();

// Legge il progetto
if (optimizeObj.LoadProject ("C:\projects\one.xml", true) == CutError.ErrorNone)
{
    //----- Settaggi di progetto
    myUnit= optimizeObj.Unit;
    myDirection= optimizeObj.Direction;
    myCorner= optimizeObj.Corner;
    myFiTool= optimizeObj.CutterDiameter;
    myTensionGap= optimizeObj.TensionGap;
    myMaxCutLevels= optimizeObj.MaxCutLevels;
    myPreCut= optimizeObj.PreCut;
    myLongCut= optimizeObj.LongCut;
    myTransvCut= optimizeObj.TransvCut;
    myZCut= optimizeObj.ZCut;
    myInfoCst[ni = 0]= optimizeObj.Custom1;
    myInfoCst[ ++ni]= optimizeObj.Custom2;
    ....
    myInfoCst[ ++ni]= optimizeObj.Custom10;

    //----- Legge i rettangoli
    for (int idxElement=0; idxElement< optimizeObj.CountRect; idxElement++)
    {
        optimizeObj.ReadRectIndex (idxElement, ref ItemRect);
        // ...
        // Lettura ed elaborazione dei dati
        //...
    }

    //----- Legge i fogli
    for (int idxElement=0; idxElement< optimizeObj.CountSheet; idxElement++)
    {
        optimizeObj.ReadSheetIndex (idxElement, ref ItemSheet);
        // ...
        // Lettura ed elaborazione dei dati
        //...
    }
}
}
```