



Job shop scheduling by simulated annealing

Thomas Bamelis & Michiel Jonckheere

KU Leuven Kulak

Academiejaar 2017-2018

Overzicht

Inleiding

De algemene vorm van simulated annealing

Waarom niet simpeler

Besluit

Overzicht

Inleiding

De algemene vorm van simulated annealing

Waarom niet simpeler

Besluit

Inleiding

Simulated annealing

Een manier om dichtbij optimale oplossing te geven voor een combinatorisch optimalisatie probleem.

Gebaseerd op het afkoelen van metalen.

Inleiding

Simulated annealing

Een manier om dichtbij optimale oplossing te geven voor een combinatorisch optimalisatie probleem.

Gebaseerd op het afkoelen van metalen.

Overzicht

Inleiding

De algemene vorm van simulated annealing

Waarom niet simpeler

Besluit

Gegevens

3 benodigdheden voor SA:

1. \mathcal{R} : Een verzameling configuraties / combinaties
2. $C : \mathcal{R} \rightarrow \mathbb{R}$: Een functie die de 'kost' van een configuratie weergeeft
3. $\mathcal{N} : \mathcal{R} \rightarrow 2^{\mathcal{R}}$: Een functie die de 'neighborhood' van een configuratie weergeeft, met $\mathcal{N}(i) \subseteq \mathcal{R}$

Een simpele transitie is nodig waaruit indien toegepast op een configuratie i , alle $\mathcal{N}(i)$ hieruit kunnen volgen

Algoritme SA

▷ Iteratief algoritme waarin $i \in \mathcal{R}$ gegeven is bij de initialisatie.

→ Er wordt een neighbor $j \in \mathcal{N}(i)$ genomen

→ De kans dat we met j verdergaan is $\min\{1, e^{\frac{-(C(j)-C(i))}{c}}\}$ met c een getal die daalt tijdens de uitvoering (cfr. *temperatuur*).

Kans daalt als:

- $C(j) - C(i)$ groot of dus j veel zwaarder
- c kleiner wordt of dus tijd vordert

Kans 1 als $C(j) \leq C(i)$

Algoritme SA

▷ Iteratief algoritme waarin $i \in \mathcal{R}$ gegeven is bij de initialisatie.

→ Er wordt een neighbor $j \in \mathcal{N}(i)$ genomen

→ De kans dat we met j verdergaan is $\min\{1, e^{\frac{-(C(j)-C(i))}{c}}\}$ met c een getal die daalt tijdens de uitvoering (cfr. *temperatuur*).

Kans daalt als:

- $C(j) - C(i)$ groot of dus j veel zwaarder
- c kleiner wordt of dus tijd vordert

Kans 1 als $C(j) \leq C(i)$

Algoritme SA

▷ Iteratief algoritme waarin $i \in \mathcal{R}$ gegeven is bij de initialisatie.

→ Er wordt een neighbor $j \in \mathcal{N}(i)$ genomen

→ De kans dat we met j verdergaan is $\min\{1, e^{\frac{-(C(j)-C(i))}{c}}\}$ met c een getal die daalt tijdens de uitvoering (cfr. *temperatuur*).

Kans daalt als:

- $C(j) - C(i)$ groot of dus j veel zwaarder
- c kleiner wordt of dus tijd vordert

Kans 1 als $C(j) \leq C(i)$

Algoritme SA

▷ Iteratief algoritme waarin $i \in \mathcal{R}$ gegeven is bij de initialisatie.

→ Er wordt een neighbor $j \in \mathcal{N}(i)$ genomen

→ De kans dat we met j verdergaan is $\min\{1, e^{\frac{-(C(j)-C(i))}{c}}\}$ met c een getal die daalt tijdens de uitvoering (cfr. *temperatuur*).

Kans daalt als:

- $C(j) - C(i)$ groot of dus j veel zwaarder
- c kleiner wordt of dus tijd vordert

Kans 1 als $C(j) \leq C(i)$

Algoritme SA

▷ Iteratief algoritme waarin $i \in \mathcal{R}$ gegeven is bij de initialisatie.

→ Er wordt een neighbor $j \in \mathcal{N}(i)$ genomen

→ De kans dat we met j verdergaan is $\min\{1, e^{\frac{-(C(j)-C(i))}{c}}\}$ met c een getal die daalt tijdens de uitvoering (cfr. *temperatuur*).

Kans daalt als:

- $C(j) - C(i)$ groot of dus j veel zwaarder
- c kleiner wordt of dus tijd vordert

Kans 1 als $C(j) \leq C(i)$

c en aantal iteraties

Kies voor c begin waarde χ_0 , eindwaarde ϵ_s en δ

$$c_{k+1} = \frac{c_k}{1 + [c_k \cdot \ln(1 + \delta) / 3 \cdot \sigma_k]}$$

met σ_k de standaard afwijking van de kost van de configuraties van de k'de iteratie

→ δ klein gekozen \Rightarrow trage 'kwalitatieve' afname c

Lengte ? van de k'de markovketen?:

$$L_k = \max_{i \in \mathcal{R}} \{|\mathcal{N}(i)|\}$$

c en aantal iteraties

Kies voor c begin waarde χ_0 , eindwaarde ϵ_s en δ

$$c_{k+1} = \frac{c_k}{1 + [c_k \cdot \ln(1 + \delta) / 3 \cdot \sigma_k]}$$

met σ_k de standaard afwijking van de kost van de configuraties van de k'de iteratie

→ δ klein gekozen \Rightarrow trage 'kwalitatieve' afname c

Lengte ? van de k'de markovketen?:

$$L_k = \max_{i \in \mathcal{R}} \{|\mathcal{N}(i)|\}$$

c en aantal iteraties

Kies voor c begin waarde χ_0 , eindwaard ϵ_s en δ

$$c_{k+1} = \frac{c_k}{1 + [c_k \cdot \ln(1 + \delta) / 3 \cdot \sigma_k]}$$

met σ_k de standaard afwijking van de kost van de configuraties van de k'de iteratie

→ δ klein gekozen \Rightarrow trage 'kwalitatieve' afname c

Lengte ? van de k'de markovketen?:

$$L_k = \max_{i \in \mathcal{R}} \{|\mathcal{N}(i)|\}$$

Kwaliteit algoritme

- ▷ Indien δ kleiner dan 0,1 gekozen wordt is de eindconfiguratie binnen 2% van het globaal minimum.

Tijdscomplexiteit = $\theta(\tau L \ln(|\mathcal{R}|))$ met τ de tijd om een nieuwe transitie door te voeren.

Overzicht

Inleiding

De algemene vorm van simulated annealing

Waarom niet simpeler

Besluit

Waarom niet simpeler?

Wat indien men met j verdergaat als $C(j) \leq C(i)$?

Stel $i \in \mathcal{R}$ en $\forall j \in \mathcal{N}(i) : C(j) \geq C(i)$
(i heeft kleinste kost van al zijn neighbors)
 \Rightarrow je zit vast in $\mathcal{N}(j)$

SA 'verplicht' om te veranderen in begin en zo niet vast te zitten

Waarom niet simpeler?

Wat indien men met j verdergaat als $C(j) \leq C(i)$?

Stel $i \in \mathcal{R}$ en $\forall j \in \mathcal{N}(i) : C(j) \geq C(i)$
(i heeft kleinste kost van al zijn neighbors)
 \Rightarrow je zit vast in $\mathcal{N}(j)$

SA 'verplicht' om te veranderen in begin en zo niet vast te zitten

Waarom niet simpeler?

Wat indien men met j verdergaat als $C(j) \leq C(i)$?

Stel $i \in \mathcal{R}$ en $\forall j \in \mathcal{N}(i) : C(j) \geq C(i)$
(i heeft kleinste kost van al zijn neighbors)
 \Rightarrow je zit vast in $\mathcal{N}(j)$

SA 'verplicht' om te veranderen in begin en zo niet vast te zitten

Overzicht

Inleiding

De algemene vorm van simulated annealing

Waarom niet simpeler

Besluit

Besluit

SA blijft relatief simpel en vermijdt de valkuil van de simpelere methode.