Case study z predmetu Evolučné algoritmy

Aplikácia simulovaného žíhania na problém obchodného cestujúceho

Ján Kľuka

Jan.Kluka@st.fmph.uniba.sk

27. januára 2000

1 Popis problému

Úloha známa ako problém obchodného cestujúceho (traveling salesman problem) je neformálne zadaná nasledovne:

> Obchodný cestujúci má navštíviť n miest (každé práve raz), ktoré sú spojené každé s každým práve jednou cestou. Prejsť rôzne cesty trvá (potenciálne) rôzne dlho. Navrhnite poradie navštívených miest tak, aby celková dĺžka cesty bola najkratšia možná.

Najprirodzenejšia formalizácia používa jazyk teórie grafov:

> Daný je kompletný n-vrcholový graf $K_n = (V,E)$, $|V| = n$, $E = \binom{V}{2}$ a ohodnotenie jeho hrán – funkcia $f : E \to \mathbf{R}^+$. Nájdite hamiltonovskú kružnicu $C = (v_1, v_2, \ldots, v_n, v_1)$, $v_i \in V$, $i \neq j$ implikuje $v_i \neq v_j$, takú, že hodnota funkcie
>
> $$F(v_1, \ldots, v_n) = f(v_n, v_1) + \sum_{i=1}^{n-1} f(v_i, v_{i+1}) \tag{1}$$
>
> je minimálna.

Tento problém patrí medzi takzvané *NP*-úplné, čoho dôsledkom je, že ho nevieme riešiť deterministicky v polynomiálnom čase. Preto je tiež ťažké rozhodnúť, či stochastický algoritmus jeho riešenia dáva správne výsledky. Vyberieme teda podtriedu tohoto problému, v ktorej vieme jednoducho charakterizovať optimálne riešenie. Takouto podtriedou je problém obchodného cestujúceho na ortogonálnej mriežke:

Kompletný graf má n^2 vrcholov, ktoré usporiadame do pravouhlej mriežky $n \times n$. Funkcia ohodnotenia f priradí danej hrane Hammingovu (manhatanskú) vzdialenosť jej koncových bodov v tejto mriežke. Ak teda v_1 je v mriežke na pozícii (x_1, y_1) a v_2 na pozícii (x_2, y_2), je $f(v_1, v_2) = |x_1 - x_2| + |y_1 - y_2|$, čo je presne dĺžka najkratšej cesty medzi týmito vrcholmi v mriežke. Dĺžka optimálnej cesty je n^2 pre párne a $n^2 + 1$ pre nepárne n.

2 Použité algoritmy

Na riešenie problému obchodného cestujúceho na ortogonálnej mriežke použijeme algoritmus *simulovaného žíhania*. Ako názov hovorí, jedná sa o abstrakciu fyzikálneho procesu používaného na odstránenie vnútorného napätia v pevných látkach. Teleso zahrejeme na dostatočne vysokú teplotu, aby sa mohlo meniť umiestnenie molekúl v jeho štruktúre a necháme ho pomaly ochladnúť. Ochladzovanie musí byť dostatočne pomalé na to, aby sa štruktúra dostala pri každej teplote do rovnovážneho stavu, ktorý je charakterizovaný Boltzmannovym rozdelením. Pravdepodobnosť, že pri teplote T je teleso v stave i s energiou E_i je v tomto rozdelení

$$w_T(E_i) = \frac{1}{Q(T)} \exp\left(-\frac{E_i}{kT}\right),$$

kde k je Boltzmannova konštanta a

$$Q(T) = \sum_i \exp\left(-\frac{E_i}{kT}\right)$$

je normalizačný faktor, v ktorom sumujeme cez všetky možné stavy i.

2.1 Základná verzia simulovaného žíhania

Na simuláciu prechodu systému častíc k rovnovážnemu stavu pri danej teplote bol navrhnutý algoritmus typu Monte Carlo, ktorý postupuje nasledovne:

Pre daný počiatočný stav $\mathbf{x}$ generuje malú poruchu tohoto stavu $\mathbf{x}'$, takú, že pravdepodobnost'zmeny $\mathbf{x}$ na $\mathbf{x}'$ je rovnaká ako pravdepodobnost'zmeny $\mathbf{x}'$ na $\mathbf{x}$. Tento nový stav je potom akceptovaný s pravdepodobnosťou

$$P(\mathbf{x} \to \mathbf{x}') = \min(1, \exp(-(E_{\mathbf{x}'} - E_{\mathbf{x}})/kT)).$$

Algoritmus a kritérium akceptovania sa nazývajú po svojom autorovi *Metropolisovym algoritmom* a *Metropolisovym kritériom*. Algoritmus vytvára nové stavy a aplikuje kritérium predpísaný početkrát (budeme ho označovať k_{max}). Platí, že stavy akceptované algoritmom pre veľké hodnoty k_{max} majú Boltzmannovo rozdelenie pre príslušnú teplotu.

Teraz môžeme žíhanie simulovať veľmi jednoducho. Pre danú počiatočnú teplotu T_{max} a počiatočný stav necháme pracovať Metropolisov algoritmus, potom teplotu vhodným spôsobom znížime a ak neklesla pod vopred určenú hranicu T_{min}, opakujeme. Najjednoduchším spôsobom znižovania teploty je jej násobenie konštantou $0 \ll \alpha < 1$.

Počas behu algoritmu môžeme sledovať zmeny veličín zodpovedajúcich štatistickou fyzikou definovaným veličinám, ktoré popisujú stav systému z makroskopického hľadiska. Sú nimi stredná hodnota energie systému, jej disperzia, entropia a tepelná kapacita.

2.2 Simulované žíhanie s elitizmom

Výsledky základného simulovaného žíhania, ktoré na niektorých problémoch nedokáže poskytnúť globálne, iba lokálne minimum, viedli k jeho vylepšovaniu. Jedným z pokusov bolo simulované žíhanie s elitizmom (ďalej tiež, kvôli jednoduchosti, elitárske simulované žíhanie), pri ktorom po zmene teploty do Metropolisovho algoritmu nevstupuje nevyhnutne stav nájdený v predchádzajúcom behu tohoto algoritmu, ale stav, ktorý dosiahol najmenšiu energiu v celej doterajšej histórii výpočtu.

Táto modifikácia stráca teoretické zázemie, pretože prestáva platiť, že stavy akceptované Metropolisovym algoritmom pre veľké k_{max} majú Boltzmannovu distribúciu pravdepodobnosti. Navyše môže mať algoritmus tendenciu nájsť pre nižšie teploty lokálne maximum rôzne od globálneho. Napriek tomu pre niektoré typy problémov poskytuje lepšie výsledky.

2.3 Paralelné simulované žíhanie

Na riešenie zložitých kombinatorických problémov bola úspešne použitá metóda paralelného simulovaného žíhania. Pri tomto prístupe prebieha súčasne žíhanie z viacerých počiatočných stavov, prehľadávame teda väčšiu časť stavového priestoru. Okrem toho s pravdepodobnosťou P_{cross} aplikujeme operáciu kríženia stavov. Táto operácia z dvoch stavov $\mathbf{x}_1$, $\mathbf{x}_2$ vytvára ich „potomkov“ $\mathbf{x}'_1$ a $\mathbf{x}'_2$. O tom, či potomkovia nahradia v množine (presnejšie multimnožine) súbežne žíhaných vektorov svojich rodičov rozhodneme na základe Metropolisovho kritéria.

Výsledkom paralelného simulovaného žíhania je stav z množiny súčasne žíhaných, ktorý má najmenšiu energiu.

Prakticky sa paralelné simulované žíhanie realizuje úpravou Metropolisovho algoritmu, pri ktorej sa v každej iterácii rozhodneme, či prebehne mutácia alebo kríženie. Mutovaný vektor (resp. krížené vektory) potom vyberáme z množiny súčasne žíhaných pomocou náhodnej premennej s rovnomerným rozdelením. Samozrejme, pri rovnakom počte iterácií k_{max} ako pri základnom simulovanom žíhaní každý jednotlivý vektor z množiny prejde menším počtom mutácií (a nahradení na základe Metropolisovho kritéria).

3 Spôsob reprezentácie problému

Je zrejmé, že riešenie problému obchodného cestujúceho je jednoznačne dané permutáciou množiny všetkých vrcholov grafu. Ak vrcholy očíslujeme $0,\ldots,n^2-1$, z čísla vrcholu k ľahko určíme jeho polohu v mriežke ako $(k \bmod n, \lfloor k/n \rfloor)$ a na základe toho jednoducho vypočítame ohodnotenie hrany medzi dvoma vrcholmi. Ekvivalentom *stavu systému* z pojmového aparátu simulovaného žíhania je v tomto prípade permutácia a *energiou stavu* je hodnota funkcie F, ako bola definovaná v (1).

Na generovanie malej zmeny stavu použijeme jednoduchý postup, v ktorom pri prechode permutáciou pri každom jej prvku s malou pravdepodobnosťou P_{mut} vymeníme tento prvok s náhodne vybratým iným prvkom permutácie.

Problém nastáva v prípade, že chceme použiť algoritmus paralelného simulovaného žíhania, pri ktorom potrebujeme implementovať kríženie permutácií. Jednoduchý spôsob, používaný na kríženie vektorov, pri ktorom zvolíme pozíciu a vymeníme časti vektorov za touto pozíciou, negeneruje z permutácií permutácie. Jedným zo spôsobov riešenia tohto problému je opraviť takýmto postupom vygenerované vektory nasledovne:

Nech $P=(p_1,\ldots,p_n)$, $Q=(q_1,\ldots,q_n)$ sú pôvodné permutácie, k je náhodne zvolené a $P'=(p_1,\ldots,p_k,q_{k+1},\ldots,q_n)$, $Q'=(q_1,\ldots,q_k,p_{k+1},\ldots,p_n)$. Zostrojíme zobrazenia $f_{PQ}: p_i \mapsto q_i$ a $f_{QP}: q_i \mapsto p_i$, kde $k < i \leq n$. Ak sa p_i pre $1 \leq i < k$ nachádza v definičnom obore zobrazenia f_{QP}, potom sa táto hodnota vo vektore P' nachádza dvakrát. Na pozícii i v P' nahradíme p_i hodnotou, ktorú nájdeme ako prvé $x_{j+1} = f_{QP}(x_j)$, pre ktoré f_{QP} nie je definovaná ($x_0 = p_i$). Na opravu Q analogicky použijeme f_{PQ}. V opravených vektoroch sa už žiadna hodnota nevyskytuje viackrát.

Inou možnosťou je reprezentovať permutáciu ľubovoľným číselným vektorom. Permutáciou, ktorú tento vektor reprezentuje je tá, ktorá ho vzostupne usporiada. Výhodou tohoto postupu je možnosť jednoduchej mutácie a kríženia vektorov, ktoré sú redukovateľné na príslušné operácie na binárnych vektoroch. Potenciálnou nevýhodou je nejednoznačnosť tejto reprezentácie, a to v oboch smeroch – existuje viac vektorov, ktoré daná permutácia usporiadava a existuje viac permutácií, ktoré usporiadajú daný vektor (pokiaľ obsahuje niektoré číslo dvakrát). Druhej nejednoznačnosti sa možno vyhnúť určením algoritmu transformácie vektora na permutáciu (ktorý vychádza z nejakého triediaceho algoritmu), prvú odstrániť nemožno. Nemenej vážna je aj nutnosť častého výpočtu permutácie z vektora (napr. pri určovaní hodnoty účelovej funkcie), ktorá vyžaduje použitie triediaceho algoritmu.

$T_{min} = 0.1$

k_{max}	2000			5000			10000		
$\alpha \setminus P_{mut}$	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
0.8	1	1	0	2	3	0	4	2	1
0.95	4	5	0	5	4	3	5	5	3
0.98	4	4	1	5	4	4	5	5	4

$T_{min} = 0.05$

k_{max}	2000			5000			10000		
$\alpha \setminus P_{mut}$	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
0.8	2	0	0	4	3	1	5	4	1
0.95	4	5	2	5	5	2	4	4	3
0.98	5	5	3	5	5	5	5	5	5

$T_{min} = 0.01$

k_{max}	2000			5000			10000		
$\alpha \setminus P_{mut}$	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
0.8	2	4	0	3	4	0	3	5	2
0.95	5	5	3	5	5	5	5	5	4
0.98	5	4	4	5	5	5	5	5	5

Tabuľka 1: Výsledky základného simulovaného žíhania pre $T_{max} = 3$.

4 Implementácia

Nebudeme sa zaoberať podrobným opisom implementácie, len načrtneme základné údaje. Implementované boli všetky tri spomínané verzie simulovaného žíhania a obe reprezentácie problému obchodného cestujúceho, pričom pri reprezentácii vektorom bol na prevod vektora na permutáciu použitý triediaci algoritmus *counting sort*, ktorý za daných podmienok pracuje v čase lineárnom vzhľadom na dĺžku vektora. Podrobnejšie informácie poskytujú komentáre v zdrojovom texte programu, ktorý bol napísaný v jazyku C++, kompilovaný v GNU C++ a po malej úprave aj v Microsoft Visual C++.

5 Testovanie a výsledky

5.1 Problém obchodného cestujúceho na ortogonálnej mriežke 5×5

V prvej fáze testovania algoritmu sme sa zamerali na jednoduchší problém, s cieľom odhadnúť vplyv jednotlivých parametrov na výsledky metódy v nádeji, že nám to umožní určiť vhodné hodnoty pre zložitejšie problémy. Metóda simulovaného žíhania je riadená veľkým počtom parametrov, preto je táto úloha dôležitá, ale zároveň aj výpočtovo veľmi náročná.

Okrem typu použitého algoritmu a reprezentácie riadia simulované žíhanie počiatočná teplota T_{max}, koncová teplota T_{min}, miera ochladzovania α, počet iterácií Metropolisovho algoritmu k_{max} a pravdepodobnosť elementárnej mutácie P_{mut}, t.j. pri prechode permutáciou pravdepodobnosť, že prvok bude vymenený s náhodne vybraným iným prvkom permutácie. V prípade paralelného simulovaného žíhania pribúdajú počet súčasne žíhaných vektorov p a pravdepodobnosť kríženia P_{mut}.

5.1.1 Základná a elitárska verzia algoritmu

Pre základnú verziu simulovaného žíhania a simulované žíhanie s elitizmom sme zvolili nasledovné hodnoty parametrov:

$T_{min} = 0.1$

k_{max}	2000			5000			10000		
$\alpha \setminus P_{mut}$	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
0.8	3	0	0	4	3	0	5	4	0
0.95	3	3	0	5	3	2	5	4	2
0.98	1	4	0	4	4	2	5	5	3

$T_{min} = 0.05$

k_{max}	2000			5000			10000		
$\alpha \setminus P_{mut}$	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
0.8	3	1	0	4	1	0	5	1	0
0.95	4	2	0	5	5	1	5	5	3
0.98	3	3	1	4	4	0	5	5	3

$T_{min} = 0.01$

k_{max}	2000			5000			10000		
$\alpha \setminus P_{mut}$	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
0.8	3	0	0	2	3	3	5	4	3
0.95	2	2	0	4	4	2	5	5	3
0.98	2	3	1	5	4	3	5	5	4

Tabuľka 2: Výsledky simulovaného žíhania s elitizmom pre $T_{max} = 3$.

T_{max}	1, 3, 5
T_{min}	0.1, 0.05, 0.01
α	0.8, 0.95, 0.98
k_{max}	2000, 5000, 10000
P_{mut}	0.05, 0.1, 0.2

Pre každú kombináciu parametrov sme vykonali 5 pokusov. Rýchlo sme zistili, že hodnota parametra T_{max} nehrá podstatnú úlohu, preto sme testy na hodnotách 1 a 5 zastavili a pokračovali len s hodnotou 3.

Podobne sme zistili, že reprezentácia pomocou všeobecného vektora dosahuje podstatne horšie výsledky ako reprezentácia priamo permutáciou. Kým, ako zakrátko uvidíme, pri reprezentácii permutáciou bolo pomerne často nájdené optimálne riešenie, výsledky algoritmu používajúceho reprezentáciu pomocou vektora nemožno nazvať v drvivej väčšine prípadov ani suboptimálnymi. Optimálne riešenie bolo nájdené len v dvoch prípadoch, pričom boli testované všetky kombinácie parametrov pre základné simulované žíhanie a žíhanie s elitizmom ako pri reprezentácii permutáciou.

Príčiny možno hľadať jednak v nejednoznačnosti reprezentácie spomínanej v časti 3 (mutácia vektora nie vždy zapríčiní zmenu reprezentovanej permutácie, pretože permutácia usporiadavajúca pôvodný vektor usporiada aj mutovaný; napr. ak zmutujeme málo významný bit niektorého z čísel), jednak v opačnom extréme – mutácia vektora spôsobí príliš radikálnu zmenu permutácie, ktorá ho usporiadava (mutácia významného bitu spôsobí, že číslo nachádzajúce sa v pôvodnom usporiadaní niekde uprostred sa dostane v novom usporiadaní na niektorú krajnú pozíciu).

Výsledky pre základné a elitárske simulované žíhanie sumarizujú tabuľky 1 a 2. Aby sme šetrili priestorom a zachovali prehľadnosť, uvádzame len počet pokusov z piatich, v ktorých algoritmy dosiahli optimum.

Z týchto tabuliek vidíme, že je nevhodné príliš zvyšovať P_{mut} — hodnota 0.2 prináša vo väčšine prípadov horšie výsledky ako nižšie pravdepodobnosti. Jednoduchým vysvetlením je, že pre veľké pravdepodobnosti mutácie pracuje algoritmus príliš náhodne a nepostupuje metódou postupného vylepšovania dosiaľ dosiahnutého výsledku. Medzi pravdepodobnosťami 0.05 a 0.1 nemožno jednoznačne rozhodnúť. Pre väčšie T_{min} sa výhodnejšou javí menšia, pre menšie T_{min} väčšia hodnota.

Ďalej sa zdá, že čím je počet iterácií Metropolisovho algoritmu k_{max} väčší, tým sú výsledky lepšie (pri týchto testoch to nie je viditeľné, ale zistíme, že neobmedzené zvyšovanie k_{max} nemá zmysel). Podobne

$T_{min} = 0.005$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01
0.8	10	5	3	1	3	4	1	4	4	3
	5	3	1	1	4	3	2	4	4	3
0.95	10	5	5	2	5	4	2	4	5	4
	5	4	4	5	4	4	4	5	4	5
0.98	10	5	4	4	5	5	5	4	5	5
	5	5	5	3	4	5	5	4	5	5

$T_{min} = 0.05$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.05	0.01	0.005	0.05	0.01	0.005	0.05	0.01	0.005
0.8	10	1	0	0	4	1	0	4	3	0
	5	2	0	0	4	3	3	4	0	3
0.95	10	5	4	2	5	5	2	5	5	3
	5	4	4	4	3	2	5	5	5	3
0.98	10	5	5	3	5	5	3	5	5	5
	5	4	5	2	5	4	3	4	4	4

$T_{min} = 0.01$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.05	0.01	0.005	0.05	0.01	0.005	0.05	0.01	0.005
0.8	10	4	3	1	2	3	0	4	1	3
	5	2	1	1	2	2	1	5	4	4
0.95	10	4	4	3	5	3	3	5	3	4
	5	5	3	2	5	3	5	5	4	4
0.98	10	5	4	3	4	5	3	4	4	3
	5	5	3	2	4	4	4	5	4	4

Tabuľka 3: Výsledky paralelného simulovaného žíhania pre $T_{max} = 3$ a $P_{cross} = 0.01$.

sa zdajú byť výhodné nižšie hodnoty T_{min}. Tieto dva parametre však zároveň predlžujú čas výpočtu.

O parametri α, poklese teploty v kroku žíhania, nedávajú výsledky jednoznačnú informáciu. Zrejme krok 0.8 je príliš hrubý, medzi krokmi 0.95 a 0.98 sa však na základe uvedených výsledkov nedá jednoznačne rozhodnúť. Okrem toho je zrejmé, že parametre α a k_{max} sa vzájomne ovplyvňujú (pri väčšom α prebiehajú za sebou nasledujúce iterácie za podobných podmienok, čo možno v podstate dosiahnuť znížením α a zvýšením k_{max}).

Pokiaľ chceme porovnať základné simulované žíhanie a simulované žíhanie s elitizmom, na základe získaných dát možno za výhodnejšiu považovať základnú verziu.

5.1.2 Paralelná verzia algoritmu

Pri testoch paralelnej verzie algoritmu sme sa už sčasti snažili využiť predchádzajúce výsledky. Znížili sme minimálnu teplotu a experimentovali s nižšími pravdepodobnosťami mutácie. Paralelné simulované žíhanie však, ako sme už spomínali, vyžaduje ďalšie dva parametre. Rozhodli sme sa zvoliť ich nasledovne (pripomeňme, že p označuje počet súbežne žíhaných vektorov):

P_{cross}	0.01, 0.05, 0.15
p	5, 10

Podobne ako pri predošlých testoch sme rýchlo zistili, že vysoká pravdepodobnosť kríženia neprináša dobré výsledky. Preto sme testy s $P_{cross} = 0.15$ prerušili. Pre zaujímavosť však uvádzame dosiahnuté hodnoty v tabuľke 5.

Hlavnú pozornosť pri pokuse o interpretáciu výsledkov sústredíme na tabuľky 3 a 4. Zbežný pohľad napovie, že aj pravdepodobnosť kríženia 0.05 je príliš veľká a $P_{cross} = 0.01$ poskytuje lepšie výsledky.

$T_{min} = 0.005$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01
0.8	10	1	4	1	3	2	1	4	5	1
	5	0	3	1	2	3	3	3	3	5
0.95	10	2	5	1	4	4	2	3	5	3
	5	2	4	4	4	5	2	4	5	4
0.98	10	3	4	2	5	4	2	5	5	3
	5	5	5	2	4	5	4	5	5	5

$T_{min} = 0.05$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.05	0.01	0.005	0.05	0.01	0.005	0.05	0.01	0.005
0.8	10	2	1	0	3	2	0	4	1	1
	5	2	1	0	3	1	0	4	1	2
0.95	10	4	2	1	2	1	2	5	3	2
	5	3	3	0	5	3	1	5	3	4
0.98	10	5	4	0	4	5	4	5	2	1
	5	4	4	3	4	5	3	4	4	4

$T_{min} = 0.01$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.05	0.01	0.005	0.05	0.01	0.005	0.05	0.01	0.005
0.8	10	2	1	0	4	2	1	4	0	0
	5	2	1	0	3	2	1	3	1	1
0.95	10	1	1	1	4	1	1	5	5	3
	5	5	1	0	5	3	1	4	2	5
0.98	10	5	4	2	4	4	2	5	4	1
	5	5	3	1	5	4	4	5	5	4

Tabuľka 4: Výsledky paralelného simulovaného žíhania pre $T_{max} = 3$ a $P_{cross} = 0.05$.

$T_{min} = 0.005$

k_{max}		2000			5000			10000		
α	$p \backslash P_{mut}$	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01
0.8	10	1	1	2	1	2	1	1	2	3
	5	2	0	0	2	2	2	2	3	1
0.95	10	1	3	0	3	1	2	1	3	3
	5	2	3	2	3	4	0	4	5	1
0.98	10	3	3	2	4	5	2	5	2	1
	5	2	3	3	4	5	2	4	5	2

Tabuľka 5: Výsledky paralelného simulovaného žíhania pre $T_{max} = 3$ a $P_{cross} = 0.15$.

T_{max}	T_{min}	α	k_{max}	P_{mut}	Dĺžka kružnice
3	0.1	0.95	10000	0.05	222, 232, 294, 234, 234
3	0.1	0.98	10000	0.05	230, 264, 198, 204, 164
3	0.05	0.95	10000	0.05	218, 224, 262, 264, 242
3	0.05	0.98	10000	0.05	254, 238, 222, 214, 248
3	0.01	0.95	10000	0.05	226, 198, 232, 222, 194
3	0.01	0.98	10000	0.05	282, 212, 210, 256, 194
3	0.01	0.96	50000	0.05	208, 234, 176
5	0.01	0.96	50000	0.05	200, 210, 194
3	0.01	0.96	50000	0.01	130, 122, 124
5	0.01	0.96	50000	0.01	134, 134, 120
3	0.01	0.97	100000	0.01	134, 122, 126
3	0.01	0.97	100000	0.001	126, 122, 116

Tabuľka 6: Dĺžka najkratšej hamiltonovskej kružnice nájdenej základným algoritmom simulovaného žíhania na mriežke 10×10

Ďalej si možno všimnúť, že nie je užitočné príliš znižovať P_{mut}, hoci údaje získané predošlými metódami naznačovali, že nižšia pravdepodobnosť mutácie pri vyššej T_{min} by mohla byť výhodná. Platí to nie len pre veľmi nízku hodnotu $P_{mut} = 0.005$, ale aj pre $P_{mut} = 0.01$.

Podobne ako v 5.1.1 môžeme konštatovať, že znižovaním T_{min} dosahujeme lepšie výsledky. Diskusiou dôvodov sme sa zaoberali spomínanom odseku.

Zložité je rozhodnúť, či je výhodnejšia menšia päťprvková, alebo väčšia desaťprvková množina súčasne žíhaných vektorov. Zrejme by si to žiadalo podrobnejší rozbor na väčšom počte pokusov. V získaných dátach vychádzajú tieto dve možnosti približne s rovnakými výsledkami a neviem vypozorovať súvislosť s inými parametrami, okrem o málo lepších výsledkov päťprvkovej množiny v tabuľke 4, t.j. pri $P_{cross} = 0.05$.

Ak porovnáme tabuľky 3 a 1, zistíme, že základné simulované žíhanie poskytuje optimálne riešenie s väčšou istotou ako paralelný variant. Pravdepodobný dôvod možno hľadať v spôsobe rozdeľovania pozornosti paralelného Metropolisovho algoritmu medzi súbežne žíhané vektory, ktorý sme spomínali v poslednom odstavci časti 2.3.

5.2 Problém obchodného cestujúceho na ortogonálnej mriežke 10×10

Pri testovaní algoritmov simulovaného žíhania na mriežke 10×10 sme nepostupovali tak systematicky ako pri testoch na mriežke 5×5. Dôvodom je výrazne väčšia výpočtová náročnosť. Ak sme teda v predošlej časti dosiahli len veľmi neurčité ohraničenia hodnôt parametrov, v tomto prípade poskytneme ešte menej všeobecné výsledky.

Hneď na začiatku testov s mriežkou 10×10 sme zistili, že parametre, ktoré umožňovali nájsť riešenie na menšom probléme takmer s istotou neposkytujú pre väčší problém ani suboptimálne riešenia. Problém na mriežke 10×10 vyžaduje o rád viac iterácií Metropolisovho algoritmu a nízku pravdepodobnosť elementárnej mutácie. Najlepšie výsledky naivných testov obsahuje prvá časť tabuľky 6.

Ako ukazuje druhá časť tejto tabuľky, len zvýšenie počtu iterácií Metropolisovho algoritmu na 50 000 na zlepšenie výsledkov nestačí. Výraznejšie zlepšenie prináša zníženie pravdepodobnosti mutácie (tretia časť) a mierny pokrok dosiahneme zvýšením počtu iterácií na 100 000 a ďalším znížením P_{mut} (štvrtá časť tabuľky 6).

Pretože sme vychádzali z predpokladu, že príčinou týchto výsledkov je nachádzanie lokálnych miním, ďalej sme pracovali s paralelnou verziou algoritmu, ktorá by pokrytím väčšej časti prehľadávaného priestoru mala nachádzať, ak nie globálne, tak aspoň najlepšie z viacerých lokálnych miním. Dĺžku minimálnych ciest nájdených týmto algoritmom za rôznych podmienok uvádza tabuľka 7.

Tabuľka 7a) nás opäť presvedča o malej úlohe počiatočnej teploty T_{max} a výhodnosti nižšej P_{mut}. Tabuľka b) ukazuje, že priveľa iterácií Metropolisovho algoritmu pri nemeniacej sa teplote nemá zmysel

a)

T_{max}	T_{min}	α	k_{max}	P_{mut}	p	P_{cross}	Dĺžka kružnice
3	0.01	0.96	150000	0.05	5	0.01	144, 126, 128
5	0.01	0.96	150000	0.05	5	0.01	134, 122, 130
3	0.01	0.96	150000	0.01	5	0.01	114, 108, 110
5	0.01	0.96	150000	0.01	5	0.01	106, 106, 108
3	0.01	0.96	150000	0.05	5	0.001	124, 114, 116
5	0.01	0.96	150000	0.05	5	0.001	128, 126, 124

b)

T_{max}	T_{min}	α	k_{max}	P_{mut}	p	P_{cross}	Dĺžka kružnice
5	0.01	0.98	150000	0.05	10	0.001	124, 122, 114
3	0.01	0.98	500000	0.01	10	0.001	110
3	0.01	0.98	500000	0.01	10	0.0001	104
3	0.01	0.98	500000	0.001	10	0.001	108
3	0.001	0.98	250000	0.01	10	0.001	102
3	0.001	0.98	250000	0.01	10	0.01	110
3	0.001	0.98	250000	0.001	10	0.001	120
3	0.001	0.98	250000	0.001	10	0.01	110

c)

T_{max}	T_{min}	α	k_{max}	P_{mut}	p	P_{cross}	Dĺžka kružnice
3	0.001	0.98	100000	0.01	10	0.001	104, 110
3	0.001	0.98	200000	0.01	10	0.001	108, 104
3	0.05	0.98	150000	0.01	10	0.001	106, 112
3	0.05	0.98	150000	0.01	20	0.001	106, 114
3	0.05	0.995	150000	0.02	25	0.001	104

Tabuľka 7: Dĺžka najkratšej hamiltonovskej kružnice nájdenej algoritmom paralelného simulovaného žíhania na mriežke 10×10

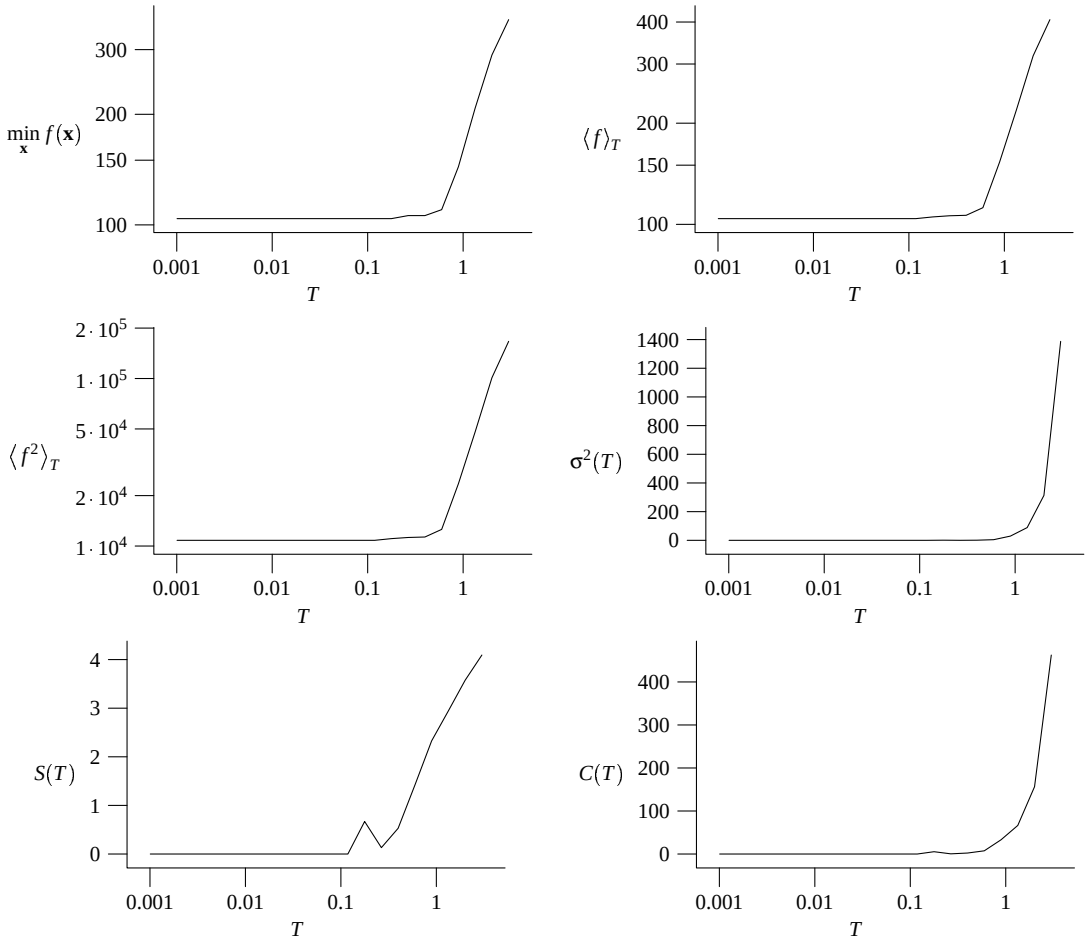
(len pre zaujímavosť – výpočty pre $k_{max} = 500000$ trvali spolu asi 12 hodín). Ako vidno z poslednej časti tejto tabuľky, najlepší výsledok sme dosiahnli znížením T_{min} na 0.001 a k_{max} na 250000 pri $P_{mut} = 0.01$ a $P_{cross} = 0.001$. Žiaľ, ako sme zistili pri pokuse o podrobnejšiu analýzu, nájdená dĺžka kružnice 102 bola skôr dielom náhody.

Základné údaje poskytuje tabuľka c). Vykonali sme ďalšie pokusy pri $T_{min} = 0.001$, pričom sme si nechali vygenerovať grafy makroskopických veličín. Tieto grafy pre prvý riadok tabuľky 7c) sú na obrázku 1, tvar cesty v niektorých časových bodoch je na obrázku 2. Z grafov je zrejmé, že po poklese teploty pod 0.1 sa už algoritmu nedarí nájsť lepšie riešenie, hoci obrázok 2 ukazuje, že aj potom sa riešenia menia.

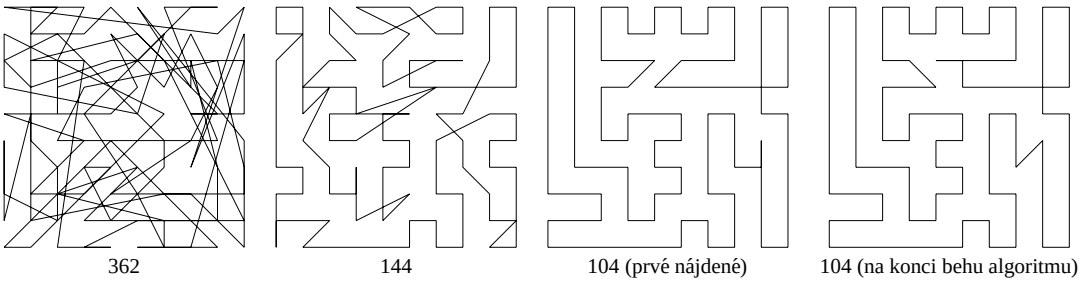
Preto sme sa rozhodli zvýšiť T_{min} na 0.05, podľa očakávania sa to neprejavilo výrazným zhoršením výsledkov (prvý riadok druhej časti tabuľky 7c)). Nakoniec sme sa ešte pokúsili pokryť väčšiu časť priestoru hamiltonovských kružníc zvýšením p na 20. Grafy pre prvý (úspešnejší) pokus s týmto nastavením sú na obrázku 3. Predovšetkým na grafe tepelnej kapacity $C(T)$ môžeme pozorovať lokálne maximum pri teplote približne 1.6. Z prednášok vieme, že takéto maximá možno považovať za indikátory javu, ktorý možno fyzikálne interpretovať ako fázový prechod.

Podľa prednášky, v takejto teplotnej oblasti je vhodné spomaliť znižovanie teploty. Naša implementácia neumožňuje dynamickú zmenu parametra α, preto sme sa pokúsili suplovať ju znížením hodnoty α na 0.995 pre celé žíhanie (čo značne predĺžilo čas potrebný na výpočet), pričom sme mierne zvýšili pravdepodobnosť mutácie (aby sme umožnili prehľadať širšie okolie súbežne žíhaných vektorov). Výsledok výpočtu s touto hodnotou ukazuje posledný riadok tabuľky 7c) a grafy na obrázku 4. Vzhľadom na pomalú zmenu teploty sme získali jemnejší priebeh makroskopických veličín a vidíme, že krivky sú nepravidelné s viacerými lokálnymi maximami.

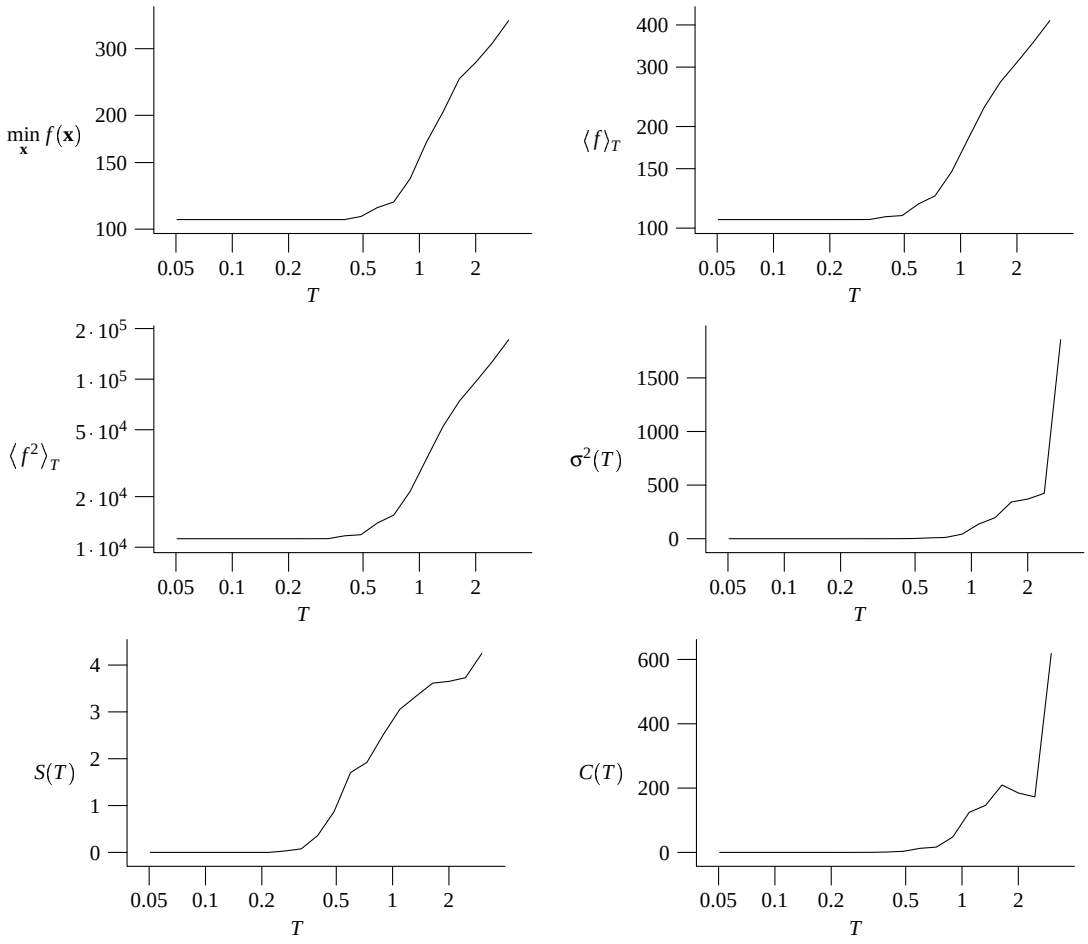
Na základe toho možno povedať, že funkcia dĺžky hamiltonovskej kružnice na ortogonálnej mriežke je veľmi zložitá s množstvom lokálnych extrémov. Naše testy ukázali, že, prinajmenšom pri tom nastavení parametrov, ktoré sme zvolili, nie je metóda simulovaného žíhania schopná poskytnúť optimálne, iba



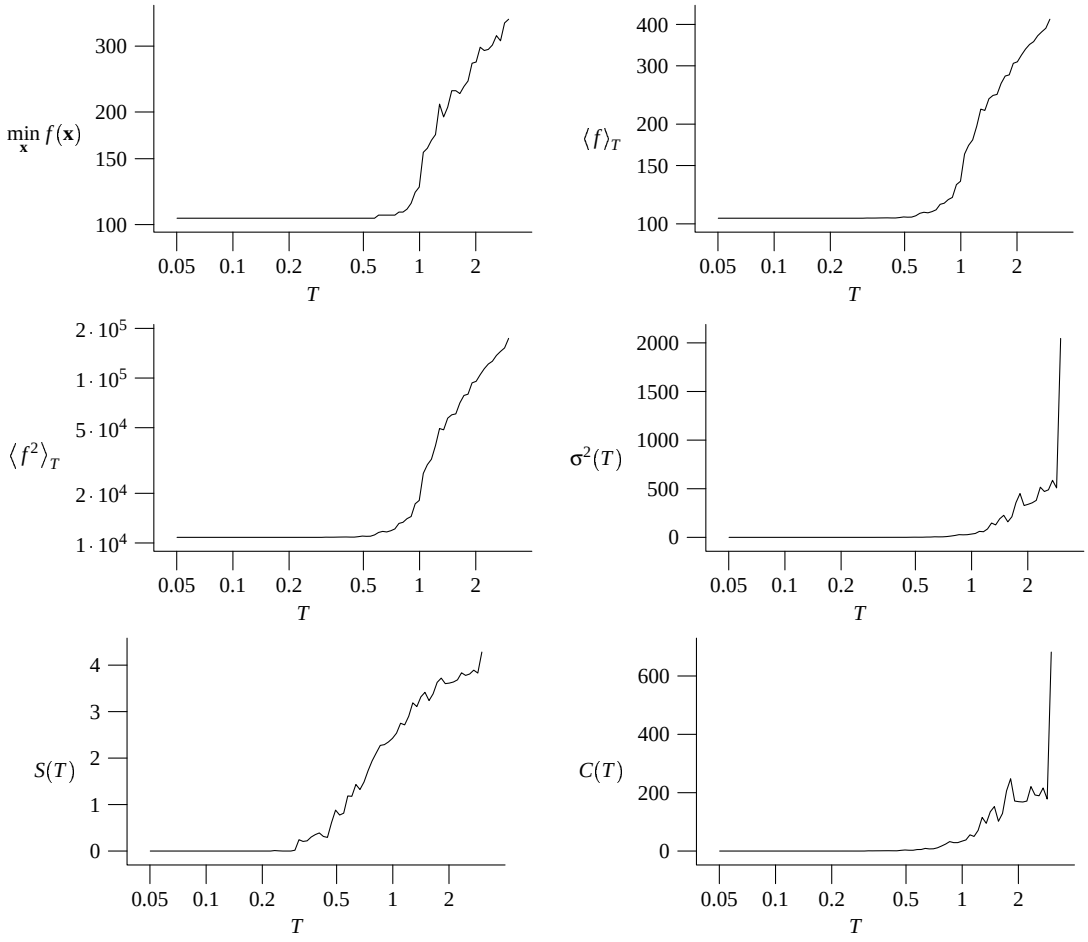
Obr. 1: Priebeh makroskopických veličín počas paralelného simulovaného žíhania za podmienok: $T_{max} = 3, T_{min} = 0.001, \alpha = 0.98, k_{max} = 100000, P_{mut} = 0.01, p = 10, P_{cross} = 0.001$ (prvý pokus, nájdená najkratšia cesta má dĺžku 104).



Obr. 2: Vývoj riešenia v pokuse na obr. 1.



Obr. 3: Priebeh makroskopických veličín počas paralelného simulovaného žíhania za podmienok: $T_{max} = 3, T_{min} = 0.05, \alpha = 0.98, k_{max} = 150000, P_{mut} = 1.01, p = 20, P_{cross} = 0.001$ (prvý pokus, nájdená najkratšia cesta má dĺžku 106).



Obr. 4: Priebeh makroskopických veličín počas paralelného simulovaného žíhania za podmienok: $T_{max} = 3, T_{min} = 0.05, \alpha = 0.995, k_{max} = 150000, P_{mut} = 0.02, p = 25, P_{cross} = 0.001$ (nájdená najkratšia cesta má dĺžku 104).

suboptimálne riešenie. Treba však mať na pamäti, že metóda je riadená veľkým počtom vzájomne sa ovplyvňujúcich parametrov a naše testy mohli byť zamerané práve na nevhodné oblasti ich hodnôt.