

文章编号: 100121595(2010)032032206

并行蚁群算法及其在区位选址中的应用

赵元, 张新长*, 康停军

中山大学 地理科学与规划学院, 广东 广州 510275

A Parallel Ant Colony Optimization Algorithm for Site Location

ZHAO Yuan, ZHANG Xinchang, KANG Tingjun

School of Geography and Planning, Sun Yat-sen University, Guangzhou, 510275, China

Abstract: An improved parallel ant colony optimization based on multiway tree is introduced to solve p2median site location problem. To take advantage of ant colony optimization and GPU parallel computing, the raster space is divided by the multiway tree and the ant paths are constructed on the nested subspace. An ideal solution can be obtained by the indirect communication of pheromone quickly. The study area is located in Guangzhou city, a densely populated region. This optimization problem considers the condition of population distribution and spatial distance. The raster layers have a resolution of 92@92 m² with a size of 512@512 pixels. A comparison experiment is conducted between the multiway tree ACO and simple search algorithms. Experiments indicate that this multiway tree ACO method can produce similar results but use lesser computation time, have better performance in convergence precision compared with the simple search algorithms. In conclusion, the proposed algorithm is important and suitable for solving site search problems.

Key words: multiway tree; ant colony optimization; parallel computation; site location; GPU general purpose computation

摘要: 提出基于多叉树并行蚁群算法的区位选址优化方法。算法依据蚁群算法具有的并行特性,采用 GPU(graphic processing unit, 图形处理器)并行运算技术,对地理空间进行多叉树划分,收集蚂蚁在多叉树层间旅行时逐步留下的信息素信息,进行路径选优获得理想的候选解,从而为解决平面空间资源优化配置问题提供新的思路。实验结果表明,与普通蚁群算法相比,采用基于多叉树搜索的并行蚁群算法,能够发挥蚁群算法的并行特征,在短时间内求得较为理想的解,适合计算大区域的空间资源配置问题。

关键词: 多叉树; 蚁群算法; 并行运算; 区位选址; GPU 通用运算

中图分类号: P208

文献标识码: A

基金项目: 国家自然科学基金(40971216)

1 引言

空间区位选址是空间规划、管理和利用中对发挥最佳效用空间位置进行遴选的基本问题。Hakimi 最早从发展次序和理论重要性方面系统阐述设施选址问题,提出中心问题与中位数问题^[1]。其后,研究者从各种角度提出不同的设施选址问题,以满足实际需要。选址问题常常面对的困难是:随着空间数据规模增加,解题规模呈现爆炸式的增加。

选址优化求解迫切需要新的解决方案。一个方向是采用启发式算法来解决空间搜索最优解问题,如:Tabari M 等采用混合式方法求解网络区位问题^[2]; Nenad M 等采用超启发式算法求解 p2 median 问题^[3]; Brooks C J、Li Xia、Fathali J 等采

用遗传算法对土地规划中的地块优化、医院选址、p2median 问题进行求解^[4]; Hifi M、Domiguez E 采用神经网络对 p2median 问题进行求解^[7,8]; Aerts C J H 等采用模拟退火算法解决资源配置问题^[9]; Bettinger P 等采用禁忌算法解决空间配置问题^[10]; He Jinqiang 等用蚁群算法解决区位配置问题^[11], Pasquale A 等采用其他启发式方法解决大规模 p2median 问题^[12]。

其中蚁群算法由 20 世纪 90 年代初意大利学者 Dorigo M 提出^[13],为模拟退火算法、遗传算法、禁忌搜索算法、人工神经网络等算法之后又一种应用于组合优化问题的启发式搜索算法。该算法是一种基于多主体模拟进化的全局搜索算法,采用分布式控制,具有自组织性和正反馈性,优化

* 为通讯作者

过程不依赖于优化问题本身的严格数学性质,具有潜在并行性。如今算法的研究已经渗透到多个应用领域,由解决一维静态优化问题也扩展到解决多维动态组合优化问题。基本蚁群算法对解决空间资源配置优化问题的启示是:对目标问题求解时,可以构造一些可能的路径,记录蚂蚁在空间上留下的信息素,根据信息素选优获得候选解。

另外一个方向是采用高性能计算手段来加快算法的效率。对于大规模或超大规模的多变量求解任务,串行算法需要大量计算时间而显得力不从心,而并行优化算法能较大幅度缩减问题求解的时间,成为近期研究热点^{[14][15]}。随着图形处理器(GPU)浮点运算能力、并行计算及可编程功能技术迅速发展,基于 GPU 通用运算得到越来越多的重视。基于 GPU 通用计算(General Purpose GPU, GPG2 PU)指利用计算机图形显卡来实现一般意义上的计算,是当前并行运算重点发展领域^{[16][17]}。并行优化算法目前应用在设施选址问题上较少,也主要限于 Open MP 并行技术进行处理^[18]。

多叉树是用于存储分级信息的数据结构,适合于对数据进行快速处理,在数据库、图像处理上有着广泛的应用。本文提出基于多叉树搜索的并行蚁群算法,利用 GPU 带来的强大并行能力,对地理空间进行多叉树分层划分;发挥多叉树分层信息搜索方式的优势,构造搜索路径,蚂蚁在多叉树的搜索路径上逐步留下信息素,借助信息素的通讯来间接协作获得理想的候选解,解决空间区位配置或选址问题。

2 多叉树蚁群算法与空间选址

2.1 空间选址问题

本文空间选址问题为:在空间 $S(w, h)$ 中进行区位选址,选址位置记为: $t(1, 2, \dots, d)$, w 和 h 分别为空间的宽度与高度, d 为选址数目;空间 S 所有点 P_{ij} 至 $p(1, 2, \dots, d)$ 的最小距离记为 d_{\min} , 求取目标函数 $f(p(1, 2, \dots, d), w, h)$ 最小值 $\min f = \sum_{i=1}^w \sum_{j=1}^h d_{\min} Q A_{ij}$ 。其中, Q 为位置 P_{ij} 的人口密度; A_{ij} 为位置 P_{ij} 的格点面积。距离计算可以采用欧几里德距离公式或采用曼哈顿距离。在实际的区域选址中,考虑到交通网特点,文中采用曼哈顿距离来表征目标位置与其他位置之间的空间移动代价。

2.2 基于多叉树划分的算法

地理空间 S 按照边均分为 N 份,每层分为

N^2 个子空间,持续划分至最小空间单元,形成 N^2 叉树的空间多叉树,最小单元大小为 $(w/N^n, h/N^n)$,层数则为 n 。所有蚂蚁从根部出发,移动的顺序分别是 0 层 y 1 层 y , $y n$ 层,最终完成对候选位置选择。若候选目标为 1,则每层要候选可能位置数目为 N^2 。若候选位置数目为 d ,则可构造 d 个 N^2 叉树。蚂蚁完成一棵 N^2 叉树后,继续向下一棵 N^2 叉树旅行,直至完成所有 N^2 叉树旅行,从而完成对多个目标的选择

现设蚂蚁分别从树的 $k-1$ 层向 k 层移动, $k \in [1, n]$ 。 $k-1$ 层位置选址记为 i , k 层的位置选址记为 j , 路径 ij 上留下的信息素记为 S_{ij}^k , 下一层要旅行的位置为 $T(d, k)$, 按照式(1)进行选择

$$T(d, k) = \begin{cases} j \operatorname{argmax} S_{ij}^k & \text{if } \operatorname{rnd} < C \\ r & \text{otherwise} \end{cases} \quad (1)$$

其中, rnd 为随机数; C 是 $[0, 1]$ 上常数; r 表示用轮盘赌策略确定下一步要旅行的子空间;选择 r 的转移概率按照式(2)确定^[19]

$$p(i, j) = \frac{S_{ij}^k}{\sum_{i=1}^{N^2} S_{ij}^k} \quad (2)$$

其中, $p(i, j)$ 表示从位置 i 转移到位置 j 的概率。

蚂蚁所选择位置在 k 层空间中的横坐标和纵坐标分别为

$$\left. \begin{aligned} T_x(d, k) &= T(d, k) \% N \\ T_y(d, k) &= T(d, k) / N \end{aligned} \right\} \quad (3)$$

其中, $\%$ 为取模运算; $/$ 为整除运算; N 为空间边的划分数。

蚂蚁在不同的层中进行选择位置过程中,路径上留下信息素。初始阶段每条路径具有信息素 S_0 。随着时间的变化,路径上的信息素要进行挥发,减少蚂蚁重复选择已有路径的概率^[19]。信息素的更新按照式(4)进行调整

$$S_{i(d, k-1), T(d, k)}^k = (1 - A) S_{i(d, k-1), T(d, k)}^{k-1} + AS_i \quad (4)$$

其中, d 为目标位置的编号; $S_{i(d, k-1), T(d, k)}^k$ 是蚂蚁从 $k-1$ 层至 k 层留下的信息素; A 表示信息素挥发因子;符号 z 表示信息素更新。

地理空间事物往往在空间邻域上存在相关性,蚂蚁选择候选位置时,候选位置的邻域状态往往也非常接近。为了增强蚂蚁对候选解邻域的感知能力,在信息素更新时,可以考虑对候选位置邻域的路径也进行信息素更新,按照式(5)进行调整 $S_{i(d, n-1), T(d, n)}^{Ne} = (1 - A) S_{i(d, n-1), T(d, n)}^{Ne-1} + AS_i$ (5) 其中, d 为候选目标编号; Ne 为候选位置 3 @ 3 范

围邻居; B 为从上一层至 Se 邻域 Ne 的信息素更新系数, A 取值区间为 $(0, 1)$; 符号 z 表示信息素更新。

当所有蚂蚁完成多叉树的旅行, 由式(6)可计算出其在空间上选择目标 d 位置

$$\left. \begin{aligned} x(d) &= \prod_{i=1}^k T_x(d, i) N^{2(k-i)} \\ y(d) &= \prod_{i=1}^k T_y(d, i) N^{2(k-i)} \end{aligned} \right\} \quad (6)$$

其中, d 为候选目标编号; $T_x(d, k)$ 为蚂蚁在 k 层子空间中的横坐标; $T_y(d, k)$ 为蚂蚁在 k 层子空间中的纵坐标。

计算每只蚂蚁对应的函数值 $f(p(1, 2, \dots, d), w, h)$, 选出最优函数值及最优蚂蚁 t_{min}

$$f_{t_{min}} = \min(f_1, f_2, \dots, f_{N_0}) \quad (7)$$

最优蚂蚁旅行路径上的信息素按式(8)作全局更新

$$S_j^k z (1 - C) S_j^k + C f(t_{min}) \quad (8)$$

其中, d 为候选目标编号; $i = T(d, k-1)$, 为蚂蚁在 $k-1$ 层选择位置; $j = T(d, k)$, 为蚂蚁在 k 层候选位置; C 为信息素调整系数; 符号 z 表示路径上信息素更新。

反复进行上面的步骤, 若达到指定的循环次数或候选解在一定循环次数后没有改进, 停止算法运行。

3 基于多叉树搜索的并行算法

从蚁群的算法描述可以看出, 算法天然具有并行性, 本质上是分布式协同优化计算方式。采用串行计算方式并不能真正体现这种本质, 研究蚁群算法的并行实现具有十分重要的意义。

3.1 蚁群算法的并行处理

3.1.1 数据分块

由于并行运算数据流宽度并非无限制, 需要将数据分块, 以满足并行运算的需要和约束。例如: 大小为 $128 @ 128$ 的地理空间可以分成 64 个大小为 $16 @ 16$ 的分块空间。如图 1 中(b)部分所示。这样, 求解整个地理空间的目标函数也被分为若干个块。块内目标函数求解是并发进行的, 结果为一组解。

3.1.2 缩减运算

缩减运算是指将一组并行数据进行多次合并求解得到最终唯一结果的过程。由于串行方法在并行环境下不能发挥并行优势效率。本文采用二

叉树归并操作方式, 将数据按照两两合并运算, 反复操作, 如图 1 中(b)部分所示。对于宽度为 n 的数据流, 缩减 \log_2^n 次数可以得到最终结果^[20]。按照并行缩减思路, 不仅可以求解并行数据的和, 同理也可以求取并行数据的差、积、最大值、最小值等类似的数值运算。

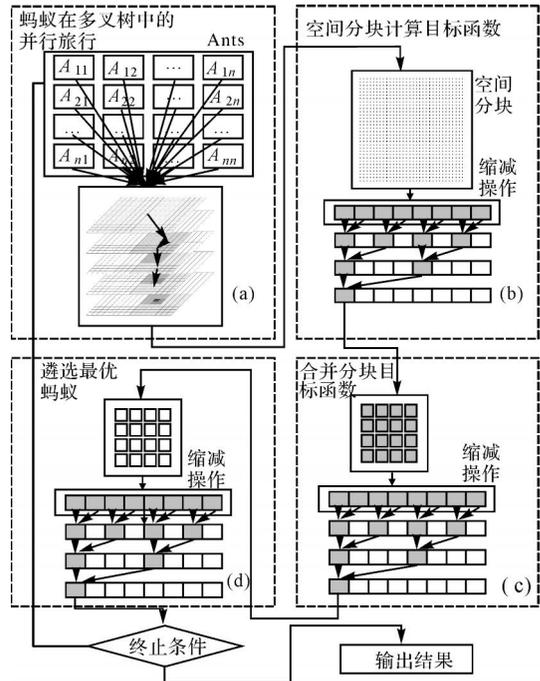


图 1 基于多叉树搜索的并行运算逻辑结构

Fig. 1 Logical structure of parallel algorithm based on multiway tree

3.2 蚁群算法并行执行步骤

对应于图 1 中并行运算的逻辑结构, 空间选址并行算法可以描述为:

第一阶段: 蚂蚁在多叉树中并行旅行(图 1 中(a)部分)

步骤 1, 构造栅格空间的 N^2 叉树结构, 初始化参数;

步骤 2, 所有蚂蚁置于 N^2 叉树的根节点;

步骤 3, 遍历 N^2 叉树, 执行步骤 4~ 步骤 8;

步骤 4, 蚂蚁并行执行步骤 5~ 步骤 6;

步骤 5, 根据式(1)~ 式(3)选择蚂蚁在 N^2 叉树的第 k 层到达的子空间, 如果所有蚂蚁遍历完所有 N^2 叉树, 则转入步骤 7;

步骤 6, 蚂蚁选择路径之后, 准备进入 N^2 叉树的下一层, 按式(4)~ 式(5), 选择下一步旅行路径, 对路径上的信息素执行局部更新;

第二阶段: 并行计算分块空间目标函数(图 1

中(b)部分)

步骤 7, 地理空间分为若干分块空间, 求出空间分块上的目标函数;

第三阶段: 并行合并分块目标函数(图 1 中(c)部分)

步骤 8, 采用缩减方法求出整个空间的目标函数;

第四阶段: 遴选最优蚂蚁(图 1 中(d)部分)

步骤 9, 采用并行缩减方法, 根据式(7)~式(8)选出最优蚂蚁, 对最优蚂蚁经过的路径上的信息素执行全局更新;

步骤 10, 判断是否满足终止条件, 若否转入步骤 2;

步骤 11, 输出结果, 停止运行。

4 案例实验

4.1 实验数据及实验环境

实验环境为单台 PC 环境, 内存配置容量为 4 GB, CPU 为双核心 Intel P4, 主频为 2.55 GHz, 图形处理器为支持 CUDA 的 NVIDIA GTS 250 显示卡。显卡拥有 16 个单指令多数据流的处理单元, 搭配 256 M 显存。操作系统环境为 Windows XP SP3, CUDA 的版本为 1.1。

实验研究区域为广州市行政辖区, 包括白云区、荔湾区、越秀区、天河区、萝岗区、黄埔区和海珠区。以公共设施的空间选址为例, 检验并行蚁群算法在基础设施空间优化布局方面的有效性。人口数据来源于广州市 2005 年 95 个街道(镇)的抽样统计数据。人口分布密度数据处理为 ASCII 栅格文本格式, 由本文所述算法进行分析。

4.2 实验分析

程序中采用的并行群算法的参数如下: $N^2=64, k=3, A=0.9, B=0.8$, 邻域半径为 3。为了适应蚁群算法, 目标函数 $f(p(1, 2, \dots, d), w, h)$ 变换为解空间区间为 $[0, 1]$ 上的函数最小化问题, 按照前述第 2 部分目标函数, 选取多个选址目标进行计算验证算法的有效性。同时采用穷举计算进行验证。经过实验, 并行算法的结果与普通算法的选址结果同直接计算是一致的, 但在计算时间上花费更少(表 1)。采用直接穷举计算方法, 选取 10 个以上的目标, 算法计算往往要耗费几十个小时以上。实际运算中, 无法应用穷举算法进行优化选址, 仅在较少选址目标情况下, 作为算法验证的手段。

表 1 并行多叉树蚁群算法耗费的时间代价

Tab. 1 Comparison of computing time for parallel computation at different spatial scale /s

空间规模	目标数目					
	2	3	4	5	10	20
64@64	17	22	29	36	66	130
128@128	68	85	115	138	257	507
256@256	277	353	438	524	982	1958
512@512	1062	1363	1711	2020	3847	7683

图 2 验证在不同目标数目下适应度与迭代次数之间的变化状况。对于选址目标函数来说, 即使适应度存在较小的差别, 对于较大区域空间中选址位置, 还是存在较大位置差别。因此, 为了寻找稳定的空间位置, 运算的次数也有所增加。图中给出的运算次数, 即是寻找到较为稳定空间位置时的情形。结果显示, 以不同的选择目标数下, 适应度均随着迭代次数增加, 能够收敛到稳定的数值, 得到较为理想的候选解。

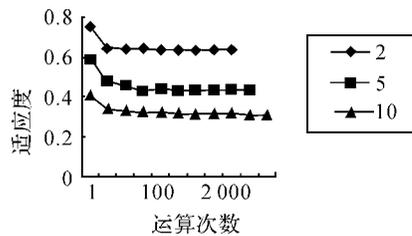


图 2 适应度收敛曲线图
Fig. 2 Convergence rate

图 3 为验证并行算法和串行算法效率对比状况。对比指标采用时间加速比来表示, 即同一算法在 CPU 下执行串行算法和 GPU 下执行并行算法的时间比率。分别选取 2~20 个选址目标, 在 GPU 并行环境下和 CPU 串行环境执行算法。结果显示, GPU 并行算法对 CPU 串行算法加速比在 7.8 至 8.8 之间。可以看出, 在目标较多和空间数据规模较大的情况下, 算法性能稍有上升, 体现出并行算法对大规模数据问题的性能潜力。

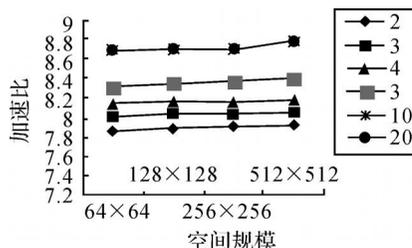


图 3 并行算法对于串行算法的加速比
Fig. 3 Speedup of parallel to serials

图4为不同选址数目情况下实验区域的运行结果。可以看出,在选址数目为5的情况下,公共设施主要集中在人口密集的越秀区和天河区。随着选址目标的增多,选址从越秀区、天河区向海珠区、白云区、萝岗区、黄埔区等空间扩散和移动。当目标选址超过20的时候,选址目标已经扩展到整个空间。选址在空间上的总体变化规律是:公共设施选址在人口密集的中心城区(越秀区和天河区)选址数量最多(要占到选址数量的一半左右),在人口密度较低的外围区大体上沿着中心城区向白云区(向北)、中心区向萝岗区(东北方向)、中心区向黄埔区(向东)、中心区向海珠区(向南)几个轴线方向进行增量扩展选址。

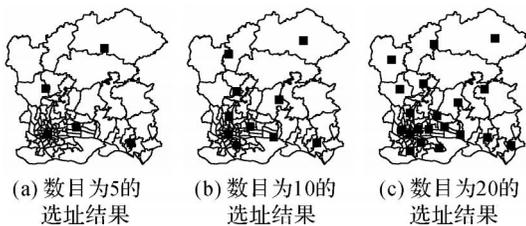


图4 不同数目选址下的运行结果

Fig. 4 Site selection for different number targets of the facility

实验表明,采用并行策略的多叉树蚁群算法与普通算法相比,具有以下优点:首先,算法建立在蚁群寻优的思想,蚂蚁选择路径时以一定的概率选择当前最好的路径^[19],由于采用概率选择策略,即使蚂蚁初始分布对应于较好的解或某个时刻陷入局部极值点上,整个蚁群也会不断地向全局最优解逼近;其次,算法采用多叉树对地理空间进行搜索,由于搜索规模增长与空间规模增长之间呈现对数关系,相对于空间规模增大,搜索规模仅呈缓慢增长趋势,在并行环境下能够发挥出并行优势,保持较快的搜索速度;最后,算法能够承受较多的蚂蚁数量(可以超过256只蚂蚁),充分发挥并行优势,获得较高的加速比。图3显示了对不同规模问题进行多次迭代时,加速比在7.8~8.8之间,大大提高了算法的运行速度。对于求解较大规模空间的选址问题来说,具有非常明显的计算时间优势。

5 结论

在地理空间数据规模较大及目标选址较多的情况下,使用普通算法求解空间选址问题很难在

空间及时间效率上得到满意解。本文提出基于多叉树搜索的并行蚁群算法,采取对栅格空间中进行多级均匀划分的策略,将蚂蚁搜索形成的信息素构建在多叉树中,蚂蚁根据多叉树中的信息搜索最好解。蚂蚁搜索过程中,对算法的初始群体不敏感,既会以一定的概率选择当前最好的路径,来保证蚂蚁沿着最好解的路径进行搜索,同时也会以一定的概率选择其他路径继续探索,从而避免蚂蚁在搜索过程中陷入局部极值。从整个蚁群搜索过程看,蚁群会不断地向全局最优解逼近。与普通蚁群算法相比,并行蚁群算法可以获得较高的加速比,尤其是在解决大区域空间的优化选址问题上,算法的效率提高更加显著。

后续研究中,准备参照现阶段蚁群算法的各种改良措施,对算法进一步优化,以求获得更好的性能。

参考文献:

- [1] HAKIMI S L. Optimum Location of Switching Centers and the Absolute Centers and Medians of a Graph[J]. *Operations Research*, 1964, 12: 452-459.
- [2] TABARI M, KABOLI A, ARYANEZHAD M B, et al. A New Method for Location Selection: a Hybrid Analysis[J]. *Applied Mathematics and Computation*, 2008, 206(2): 5982-606.
- [3] NENAD M, JACK B P H, JOSE A M. The p2median Problem: a Survey of Metaheuristic Approaches [J]. *European Journal of Operational Research*, 2007, 179: 927-939.
- [4] BROOKES C J. A Genetic Algorithm for Designing Optimal Patch Configurations in GIS[J]. *International Journal of Geographical Information Science*, 2001, 15(6): 532-559.
- [5] LI Xia, YEH A G O. Optimal Spatial Search Using Genetic Algorithms and GIS[J]. *Acta Geographica Sinica*, 2004, 59(5): 745-753. (黎夏, 叶嘉安. 遗传算法和GIS结合进行空间优化决策的研究[J]. *地理学报*, 2004, 59(5): 745-753.)
- [6] FATHALI J. A Genetic Algorithm for the p2median Problem with pos/neg Weights [J]. *Applied Mathematics and Computation*, 2006, 183: 1072-1083.
- [7] HIFI M, PAXHOS V T, ZISSIMOPOULOS V. A Neural Network for Minimum Set Problem [J]. *Chaos, Solitons and Fractals*, 2000, 11: 2072-2089.
- [8] DOMINGUEZ E, MUSOZA J. A Neural Model for the p2median Problem [J]. *Computers & Operations Research*, 2008, 35: 402-416.
- [9] AERTS C J H, HEUVELINK G B M. Using Simulated Annealing for Resource Allocation [J]. *International*

- Journal of Geographical Information Science, 2002, 16(6): 572-587.
- [10] BETTINGER P, SESSIONS J, BOSTON K. Using Tabu Search to Schedule Timber Harvests Subject to Spatial Wildlife Goals for Big Game [J]. Ecological Modeling, 1997, 94: 111-123.
- [11] HE Jinqiang, LI Xia, LIU Xiaoping, et al. Ant Colony Algorithms for Optimal Site Selection in Large Regions [J]. Journal of Remote Sensing, 2009, 13(2): 246-255. (何晋强, 黎夏, 刘小平, 等. 蚁群智能及其在大区域基础设施选址中的应用 [J]. 遥感学报, 2009, 13(2): 246-255.)
- [12] PASQUALE A, ANTONIO S, IGOR V. A Heuristic for Large-scale p2median Instances [J]. Electronic Notes in Discrete Mathematics, 2003, 13(2): 142-17.
- [13] DORIGO M, MANIEZZO V, COLORNI A. Ant System: Optimization by a Colony of Cooperating Agents [J]. IEEE Transactions on Systems, Man and Cybernetics, 1996, 26(1): 29-41.
- [14] GENDRON B. Parallel Computing in Combinatorial Optimization [M]. Pisa: [s. n.], 2005.
- [15] DELISLE P, KRAJECKI M, GRAVEL M, et al. Parallel Implementation of an Ant Colony Optimization Metaheuristic with OpenMP [C]. Proceedings of the 3rd European Workshop on OpenMP (EWOMP. 01). Barcelona: [s. n.], 2001.
- [16] OWENS J D, LUEBKE D, GOVINDARAJU N. A Survey of General Purpose Computation on Graphics Hardware [C]. MEuroGraphics 2005. Dublin: [s. n.], 2005: 212-51.
- [17] CHITTY D M. A Data Parallel Approach to Genetic Programming Using Programmable Graphics Hardware [C]. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. New York: ACM, 2007: 1562-1573.
- [18] FELIX G L, BELEN M B, JOSE A M P, et al. Parallelization of the Scatter Search for the p2median Problem [J]. Parallel Computing, 2003, 29(5): 575 - 589.
- [19] CHEN Ye. Ant Colony Optimization Algorithm to Continuous Function [J]. Journal of Sichuan University: Engineering Science Edition, 2004, 36(6): 117-120. (陈焯. 用于连续函数优化的蚁群算法 [J]. 四川大学学报: 工程科学版, 2004, 36(6): 117-120.)
- [20] PHARR M. GPU Gems 2 [M]. [S. l.]: Addison-Wesley, 2005: 70-705.

(责任编辑:丛树平)

收稿日期: 2010-02-19

修回日期: 2010-03-15

第一作者简介: 赵元(1977), 男, 博士生, 研究方向为地理信息建模与土地利用时空结构演变。

First author: ZHAO Yuan (1977), male, PhD candidate, majors in geographic information modeling and spatio-temporal structure change of land use.

E-mail: giszy@163.com

(上接第321页)

- Lecture Notes in Computer Science, 2005, 3633: 732-90.
- [14] HU H, LEE D, XU J. Fast Nearest Neighbor Search on Road Networks [J]. Lecture Notes in Computer Science, 2006, 3896: 186-203.
- [15] HUANG X, JENSEN C S, SALTENIS S. GRID: A Versatile Approach to Efficient Query Processing in Spatial Networks [J]. Lecture Notes in Computer Science, 2007, 4605: 932-111.
- [16] KIRBY R F, POTTS R B. The Minimum Route Problem for Networks with Turn Penalties and Prohibitions [J]. Transportation Research, 1969, 3(3): 39-48.
- [17] BRINKHOFF T, STRÖM. A Framework for Generating Network-based Moving Objects [J]. Geoinformatica, 2002, 6(2): 153-180.
- [18] CIVILIS A, JENSEN C S, PAKALNIS S. Techniques for Efficient Road-network-based Tracking of Moving Objects [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(5): 692-712.

(责任编辑:雷秀丽)

收稿日期: 2008-10-24

修回日期: 2009-02-25

第一作者简介: 许林(1981), 男, 博士生, 现从事移动对象的位置管理、车辆导航、位置服务等研究。

First author: XU Lin (1981), male, PhD candidate, majors in moving object location management, vehicle navigation and LBS.

E-mail: xuling717@126.com