

一种全新的0-1背包问题的优化方法*

史 岚, 吕建辉

(东北大学 信息科学与工程学院, 沈阳 110819)

摘要: 为了进一步优化难解背包问题, 在传统理论上给出了一种基于动态预期效率的经济学模型, 构造了一种全新的背包优化算法, 并进行了单独仿真实验和对比实验仿真。实验表明, 在同一类背包问题中, 该算法优于贪心算法、回溯法、动态规划算法和分支限界算法; 与萤火虫群算法对比, 该算法较大幅度地提高了收敛速度并节省了存储空间, 收敛速度几乎是萤火虫群算法的10倍。最后, 经过对20个背包问题的探究, 验证了该算法的可行性, 并确定了该算法的适应范围。

关键词: 0-1背包; 动态预期效率; 收敛速度; 存储空间; 适应范围

中图分类号: TP18; TP301.6

文献标志码: A

文章编号: 1001-3695(2014)04-0997-04

doi:10.3969/j.issn.1001-3695.2014.04.009

New optimization algorithm of 0-1 knapsack problem

SHI Lan, LV Jian-hui

(College of Information Science & Engineering, Northeastern University, Shenyang 110819, China)

Abstract: In order to do further research on enigmatical knapsack problems, this paper proposed an economic model based on dynamic expectation efficiency, and established a new optimization algorithm of 0-1 knapsack problem after analysis and research with the traditional theory of solving knapsack problem. And this paper gave the individual experiment and comparison experiment with artificial glowworm swarm algorithm. The results of experiment show that the algorithm is better than greedy algorithm, backtracking algorithm, dynamic programming algorithm and bound algorithm in the same 0-1 knapsack problem. In comparison with artificial glowworm swarm algorithm, this algorithm improves convergence speed largely and saves the storage space, and the convergence speed is ten times as the artificial glowworm swarm algorithm. Finally, this paper gave 20 0-1 knapsack problems, proved the feasibility of the algorithm, and determined an adaptive scope of the algorithm.

Key words: 0-1 knapsack; dynamic expectation efficiency; convergence speed; storage space; adaptive scope

背包问题属于难解的NP^[1]完全问题。近年来,许多国内外专家、学者对其进行了深入的研究,现有的求解方法如贪心算法、分支限界算法、回溯法、动态规划算法、e近似算法^[2-6]以及改进的蜂群算法、蚁群算法、萤火虫群算法^[7-9]等,这些算法都是基于数学原理进行展开的。由于各个算法中的数学模型不同,算法的复杂性和收敛速度也各有不同。事实上,如何提高算法的收敛速度才是研究的主要思路,纵观现存的这些求解算法,其收敛速度仍有提升的空间。

0-1背包问题是不可分割的背包问题,具体描述如下:在0-1背包问题中,物体或者被装入背包,或者不被装入背包,只有两种选择。假设取 $x_i(0,1)$ 表示物体 i 被装入背包的状态。当 $x_i=0$ 时,表示物体没有装入背包;否则表示装入背包。根据问题的要求,有下面的约束方程和目标函数:

$$\sum_{i=1}^n w_i x_i \leq M \quad (1)$$

$$\text{optp} = \max \sum_{i=1}^n c_i x_i \quad (2)$$

其中: M 为背包载重量; w_i 为各个物体的重量; c_i 为各个物体的价值。于是,问题归结为寻找一个满足上述约束方程并使目标函数达到最大的解向量 $X=(x_1, x_2, \dots, x_n)$ 。

在文献[10]中提到了静态预期效率的算法思想,并在结论部分提出了基于动态模型的预期效率的设想。本文将探讨

动态的 $\text{optp}(i)$ 对算法的影响。

1 算法基本思想

1.1 算法思想决策

a)贪心策略在解决局部最优解时有较好的适用度。选择前 $n/2$ 个待选物体首先装入背包,如果当前装入背包物体的总重量超出背包的载重量,那么重新装入前 $n/4$ 个待选物体,再次判断当前装入背包物体的总重量是否超出背包的载重量。如果成立,继续二分;否则,对剩余待选物体进行预期。

b)贪心策略之后,背包的剩余载重量和当前装入背包的价值总量是重点关注的对象,预期的目的是使剩余载重量快速收敛到0,且价值总量快速达到更大。

1.2 数学模型的建立

静态预期效率模型如式(3)所示。

$$f_i(M, n, w, c, r) = \frac{r_i}{r_{i-1}} \left| \frac{(\text{optp}(i) - \sum_{j=1}^{\lfloor n/2 \rfloor} c_j - \sum_{j=\lfloor n/2 \rfloor + 1}^i c_{j-1} x_{j-1}) - (n-i+1)c_i}{(M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor + 1}^i w_{j-1} x_{j-1}) - (n-i+1)w_i} \right| \quad (3)$$

其中,目标函数值是式(3)中的一个未知元。为了方便模型的迭代,设置 optp 各处相等,如式(4)所示。

收稿日期:2013-06-29;修回日期:2013-08-12 基金项目:国家自然科学基金资助项目(61100182)

作者简介:史岚(1964-),女,辽宁沈阳人,副教授,硕士,主要研究方向为计算机网络与信息安全(shilan@mail.neu.edu.cn);吕建辉(1988-),男,硕士研究生,主要研究方向为计算机网络与信息安全。

$$\text{optp}(\lfloor n/2 \rfloor + 1) = \dots = \text{optp}(i) = \dots = \text{optp}(n - \lfloor n/2 \rfloor) \quad (4)$$

要想把式(3)演变成动态预期模型,必然要修改条件模型式(4),即不能让 $\text{optp}(i)$ 单纯地相等,而是根据各个物体的情况以及上一个物体的情况进行预期。

使剩余载重量快速收敛到 0,要在剩余待选物体中抽取有效的物体,这必须要满足:每装入一个物体,由于剩余载重量的缩减,则接下来期望得到的价值也会减小。一般有两种期望,一种是基于整体装入背包的物体重量和价值比的期望,另一种是基于临近待装入背包的物体重量和价值的期望,后一种也叫按照上一个物体的性价比进行期望。这两种期望模型如式(5)所示。

$$\begin{cases} \frac{\text{optp}'(i)}{M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}} = \frac{\sum_{j=1}^{\lfloor n/2 \rfloor} c_j + \sum_{j=\lfloor n/2 \rfloor+1}^i c_{j-1} x_{j-1}}{\sum_{j=1}^{\lfloor n/2 \rfloor} w_j + \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}} & (5a) \\ \frac{\text{optp}'(i)}{M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}} = \frac{c_{i-1}}{w_{i-1}} & (5b) \end{cases}$$

其中: $\text{optp}'(i)$ 表示对背包剩余载重量的期望装入价值。很显然,在同一时刻,用式(5a)期望出来的 $\text{optp}'(i)$ 要高于用式(5b)期望出来的 $\text{optp}'(i)$ 。笔者在解决实际背包问题时发现,由于用式(5a)对剩余背包载重量期望装入的价值过高,排在前面物体的期望效率总高于排在后面物体的期望效率,逐渐演变成了具有贪心性质的模型。因此,本文选用式(5b)作为剩余期望模型。

式(3)中的 $\text{optp}(i)$ 表示如下:

$$\text{optp}(i) = \text{optp}'(i) + \sum_{j=1}^{\lfloor n/2 \rfloor} c_j + \sum_{j=\lfloor n/2 \rfloor+1}^i c_{j-1} x_{j-1} \quad (6)$$

将式(6)和式(5b)代入到式(3)中,经过化简得到动态预期模型,如式(7)所示。

$$f_i(M, n, w, c, r) = \frac{r_i}{r_{i-1}} \left| \frac{r_{i-1} (M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}) - (n-i+1) c_i}{(M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}) - (n-i+1) w_i} \right| \quad (7)$$

针对整体算法的操作给出以下四点说明:

a) 动态模型运算结束条件。通常情况下,当 $M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1} = 0$, 背包剩余载重量收敛到 0。然而,当前装入背包中的所有物体总重量并不是恰好等于背包的载重量,因此还要寻找另外的方法作为动态模型的结束条件。动态预期效率模型分为两个部分,第一部分是单独预期,即 r_i/r_{i-1} , 第二部分是均衡预期,即式(7)的绝对值部分。其中 $(n-i+1)$ 是为了均衡剩余量与当前量而设定的,故当 $(M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}) / (n-i+1)$ 趋于 0 时, $(n-i+1)$ 已经失去了均衡的作用,此时视为模型运算结束。综上所述,设定动态模型运算的结束条件为

$$0 \leq (M - \sum_{j=1}^{\lfloor n/2 \rfloor} w_j - \sum_{j=\lfloor n/2 \rfloor+1}^i w_{j-1} x_{j-1}) / (n-i+1) < \varepsilon, \varepsilon \rightarrow 0 \quad (8)$$

b) 算法结束条件。动态模型运算结束,对各个存在的 $f_i(M, n, w, c, r)$ 进行大小排序,累加对应的物体价值,直到背包再也装不下任何一个物体为止。

c) 绝对贪心预期出来的价值的均值总大于当前物体的价值,为了避免这种情况出现,算法规定:把式(7)中绝对值部分的分子大于 0 时的预期效率首先作为以后踢出(需要踢出的

情况下)的对象,即累加物体的价值时,不再存在这样的 $f_i(M, n, w, c, r)$ 。

d) 绝对值的意义。当动态模型运算到某个时刻,虽然预期价值的均值不超过当前物体的价值,但存在预期重量的均值超过当前物体的重量,这种现象是合理的,为了保证 $f_i(M, n, w, c, r)$ 值的非负性,要对式(7)的后半部分添加绝对值。

2 算法描述与分析

2.1 算法描述与流程

基于动态预期效率算法(dynamic expectation efficiency algorithm, DEE)的 0-1 背包优化算法,具体描述如下:

- 初始化背包的剩余载重量为 M , 背包价值量为 0。
- 把物体按性价比从高到低进行排列。
- 按照 1.1 节中算法思想 a) 提出的贪心策略装入物体, 修改剩余背包载重量和背包价值量。
- 按照算法思想 b) 提出的动态预期效率计算, 根据式(7)计算出各个 $f_i(M, n, w, c, r)$, 判断式(7)说明步骤 c) 是否成立, 成立设置 $x_i = 0$; 否则转步骤 e)。
- 判断式(7)说明算法思想 a) 是否成立, 成立转步骤 f); 否则, 转步骤 d)。
- 判断式(7)说明算法思想 b) 是否成立, 成立转步骤 g); 否则, 继续装入较高 $f_i(M, n, w, c, r)$ 的物体。
- 计算出 optp , 修改剩余背包载重量和背包价值量。
- 结束整个算法。

动态预期效率算法的流程如图 1 所示。

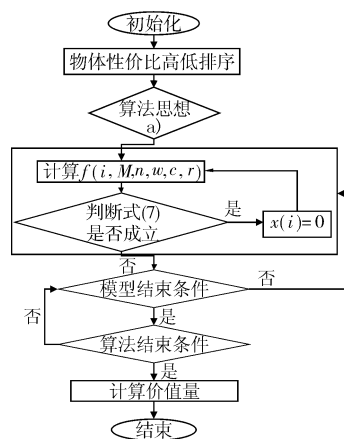


图 1 DEE 算法流程

2.2 算法分析

1) 时间复杂度分析 整个算法主要有六个部分消耗时间, 第一部分是物体性价比的排序, 第二部分是算法思想 a) 中的贪心策略, 第三部分是动态模型的计算, 第四部分是模型运算过程中条件的判断, 第五部分是现存 $f_i(M, n, w, c, r)$ 的排序, 第六部分是符合要求的待选物体的价值量累加。六个部分执行的都是基本算法^[11], 时间复杂度都是 $O(n)$, 因此整个算法的时间复杂度是 $O(n)$ 。算法的迭代次数由算法思想 a) b) 两个部分进行运算。

2) 空间复杂度分析 贪心策略部分, 每装入一个物体都要有一个 x_i 记述, 故开辟的空间为 $n/2$ 。动态预期部分, 每计算一个预期效率也要开辟一个空间去存储对应的 $f_i(M, n, w, c, r)$ 值, 只有踢出一个较小的 $f_i(M, n, w, c, r)$ 时才能节省一个

存储空间,否则要继续开辟。因此空间复杂度为 $O(n)$ 。

3 实验仿真

为了验证算法的合理性和适应性,对算法的整体思想运用 VC 6.0 进行了程序的编写,并在 Intel Core2 Q8400 2.66 GHz CPU、4 GB 内存的 Windows7 环境下运行。主要进行了两个方面的实验,分别是算法的单独检验并通过离散系数^[12]分析、与萤火虫群 (artificial glowworm swarm optimization, GSO) 的对比实验。

3.1 单独仿真实验

这部分给了两个算例 A 和 B,其中 A 的三种数据的离散系数(物体的重量离散系数、物体的价值离散系数、物体的性价比离散系数)有两个处于同一水平,B 的三种数据的离散系数各不处于同一水平。

A: $n = 20, M = 550, w = \{45, 27, 58, 25, 33, 42, 55, 62, 19, 72, 80, 24, 59, 35, 14, 66, 45, 83, 71, 40\}, v = \{93, 38, 90, 30, 56, 43, 49, 77, 36, 99, 120, 70, 48, 70, 38, 89, 29, 94, 62, 80\}$ 。三种数据的离散系数分别是 0.4239、0.3943、0.3955,可见物体的价值离散系数与物体的性价比离散系数处于同一水平。回溯法、动态规划算法、分支限界法求得最好解为 909,贪心算法的最好解为 879。运用静态预期模型求得最优解为 909,迭代 18 次;运用动态预期模型求得最优解为 909,迭代 23 次。这种情况下,动态模型与静态模型的比较如图 2 所示。

从图 2 中可以看出,动态预期模型与静态预期模型在迭代到第 15 次之前,目标函数值相同;第 15 次迭代到第 20 次的迭代过程中,采用动态预期模型暂时不影响目标函数值,之后每

历经一次迭代就会改变目标函数值。总体看来,两模型的迭代次数相差不大,能够接受。

B: $n = 20, M = 950, w = \{9, 36, 67, 29, 58, 19, 98, 205, 39, 96, 45, 79, 49, 88, 124, 105, 38, 113, 300, 160\}, v = \{26, 70, 130, 50, 93, 29, 145, 300, 56, 137, 62, 108, 65, 116, 160, 120, 40, 100, 200, 82\}$ 。三种数据的离散系数分别是 0.7765、0.6082、0.4367,没有任意两个处于同一水平。回溯法、动态规划算法、分支限界法求得最好解为 1420,贪心算法最好解为 1387。运用静态预期模型求得最优解为 1387,迭代 19 次;运用动态预期模型求得最好解为 1420,迭代 19 次。这种情况下,动态模型与静态模型的比较如图 3 所示。

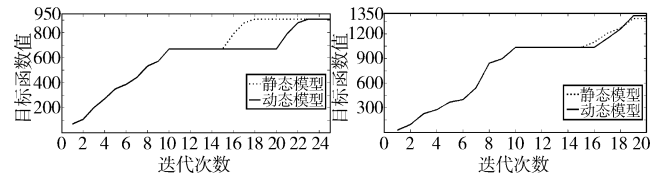


图2 例 A 情况下动态模型与静态模型的对比

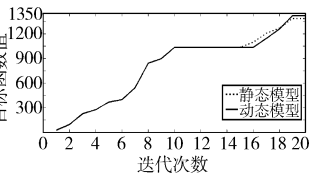


图3 例 B 情况下动态模型与静态模型的对比

很显然,静态预期模型未能求得最好解。在迭代到 15 次之前,两模型的目标函数值的走向一致,这说明满足了局部贪心最好解的性质。动态预期模型从第 16 次迭代到第 19 次,目标函数值再次连续变化。针对三种离散系数均不在同一水平的背包问题,显然动态预期模型要优于静态预期模型。

为了进一步研究算法的应用范围,给出了 20 个背包问题^[13]。待选物体个数设置为 20,背包的载重量设置为 500 ~ 950,结果如表 1 所示。

表1 三种数据的离散系数对动态模型下的最优解的影响

参数		1	2	3	4	5	6	7	8	9	10
离散系数	w	0.732 5	0.857 6	0.776 5	0.490 1	0.737 9	0.534 8	0.539 1	0.514 9	0.342 6	0.499 3
	c	0.655 2	0.637 1	0.608 2	0.439 1	0.468 3	0.465 5	0.529 6	0.523 9	0.324 5	0.477 9
	r	0.792 5	0.709 6	0.436 7	0.356 2	0.332 8	0.521 9	0.541 7	0.337 3	0.331 3	0.055 7
所求解		1 045	1 282	1 420	1 059	1 395	1 198	790	613	848	632
最优解		1 045	1 284	1 420	1 059	1 407	1 198	790	613	848	632
解差距		0	2	0	0	12	0	0	0	0	0
参数		11	12	13	14	15	16	17	18	19	20
离散系数	w	0.514 8	0.542 4	0.637 4	0.540 3	0.623 2	0.484 2	0.606 8	0.631 5	0.423 9	0.542 7
	c	0.467 2	0.495 2	0.531 9	0.445 3	0.525 9	0.455 2	0.461 8	0.606 5	0.394 3	0.481 0
	r	0.425 4	0.352 3	1.479 6	0.469 7	0.730 5	0.516 9	0.474 4	1.058 9	0.395 5	1.011 8
所求解		973	843	783	1 062	709	833	969	848	909	1 009
最优解		973	843	821	1 062	709	833	969	848	909	1 024
解差距		0	0	38	0	0	0	0	0	0	15

通过表 1 可以看出动态预期模型的适用范围:三种数据的离散系数任意两个处于同一水平的所有情况,三种数据的离散系数小于 1 且都不处于同一水平的部分情况。对于三种数据离散系数其中一个大于 1、其余两个小于 1 且不在同一水平的情况是动态预期效率模型的盲区。

3.2 与基于萤火虫群算法的对比实验

这部分的实验数据来源于文献[14]中的例 1 和例 3,两者就算法的迭代次数、最好解以及萤火虫数/开辟空间数进行了对比,结果如表 2 所示。

表2 动态预期算法与萤火虫群算法的对比结果

算法	例 1			例 3		
	最大迭代次数	最优解	萤火虫数/开辟空间数	最大迭代次数	最优解	萤火虫数/开辟空间数
GSO	100	295	30	400	3 103	100
DEE	11	295	7	42	3 103	27

萤火虫群算法和动态预期效率算法的目标函数值与迭代次数的关系如图 4、5 所示。

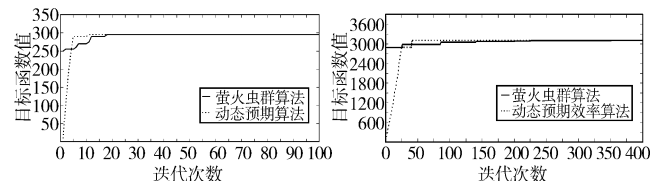


图4 例 1 中 GSO 算法和 DEE 算法的收敛情况

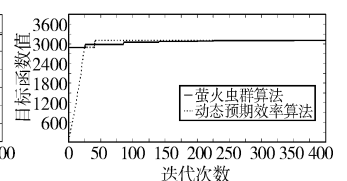


图5 例 3 中 GSO 算法和 DEE 算法的收敛情况

由图 4 看出,第 5 次迭代之前,DEE 目标函数值的收敛速度明显快于 GSO 目标函数值的收敛速度。到第 11 次迭代,DEE 算法求得最优值为 295,然而此时 GSO 算法求得的值并不是最优的。

由图 5 看出,第 25 次迭代之前,DEE 目标函数值的收敛速度明显快于 GSO 目标函数值的收敛速度。到第 27 次迭代,

DEE 算法求得最优值为 3103, 然而此时 GSO 算法求得的值并不是最优的, 需要进一步的迭代。

图 4、5 中虚线代表的是 DEE 算法, 直线上升部分是贪心策略阶段, 后期平稳上升部分是动态预期阶段。尽管 GSO 算法求得的初始目标函数值较大, 然而在后期的迭代过程中, 目标函数值的变化比较缓慢。综合而言, DEE 算法的平均收敛速度几乎是 GSO 算法的 10 倍, 在空间的开辟上几乎是 GSO 算法的 1/4。

4 结束语

动态预期效率模型是针对解决 0-1 背包问题提出的一种全新模型, 其简单实用, 不需要像静态预期效率模型那样复杂的迭代运算。与静态模型相比, 解决了两个问题: a) 三种数据的离散系数, 任意两种处于同一水平的任意情况; b) 三种数据的离散系数, 互不处于同一水平且各个离散系数小于 1 的某些问题。与萤火虫群算法对比, 动态预期效率算法具有收敛性快的优势, 并且开辟的存储空间也明显少于萤火虫的个数。总体看来, 本文算法从深层次上解决了一大类背包问题, 能广泛应用于实际问题中。在本文完成之际, 笔者大胆提出另一种数学模型, 即五个属性之间存在一种隐含关系, 采用多目标因素的线性回归模型进行解决, 这将在以后的工作中继续研究和发展。

参考文献:

- [1] RAZBOROV A A. On provably disjoint NP-pairs, RS-94-36 [R]//Basic Research; in Computer Science. 1994: 1-26.
- [2] 饶卫振, 金淳, 陆林涛. 考虑边位置信息的求解 ETSP 问题改进贪婪算法[J]. 计算机学报, 2013, 36(4): 836-850.
- [3] MASLOV E, BATSYN M, PARDALOS P M. Speeding up branch and bound algorithms for solving the maximum clique problem[J]. Journal of Global Optimization, 2013, 57(5): 1-12.
- [4] FOUCAL H, HABBAS Z. Distributed backtracking algorithm based on tree decomposition over wireless sensor networks[J]. Concurrency Computation: Practice and Experience, 2013, 25(5): 728-742.
- [5] LANTOINE G, RUSSELL R P. A hybrid differential dynamic programming algorithm for constrained optimal control problems[J]. Journal of Optimization Theory and Applications, 2012, 154(2): 418-423.
- [6] NUTOV Z, BENIAMINY I, YUSTER R. A $(1-1/e)$ -approximation algorithm for the generalized assignment problem[J]. Operations Research Letters, 2006, 34(3): 283-288.
- [7] ZHOU Fei-hong, LIAO Zi-zhen. A particle swarm optimization algorithm[J]. Applied Mechanics and Materials, 2013, 303-306: 1369-1372.
- [8] YOO K S, HAN S Y. A modified ant colony optimization algorithm for dynamic topology optimization[J]. Computers and Structures, 2013, 123(7): 68-78.
- [9] BHUSHAN B, PILLAI S S. Particle swarm optimization and firefly algorithm; performance analysis[C]//Proc of the 3rd IEEE International Advance Computing Conference. 2012: 746-751.
- [10] 史岚, 张义宏, 吕建辉. 基于绝对贪心和预期效率的 0-1 背包问题优化[J]. 计算机应用研究, 2014, 31(3): 684-687.
- [11] NAKAMURA R, TADA A, ITOKAWA T. An approximate analysis of the AVL balanced tree insertion algorithm[J]. Trans of the Information Processing Society of Japan, 1998, 39(4): 1006-1013.
- [12] YAN Zhen-ya. Exact solutions of nonlinear dispersive $K(m, n)$ model with variable coefficients[J]. Applied Mathematics and Computation, 2011, 217(22): 9474-9479.
- [13] LV Jian-hui. The experiment data on 0-1 knapsack problem[EB/OL]. (2013-06-18). <http://user.qzone.qq.com/1020052739/infocenter#!app=2&via=QQ.HashRefresh&pos=add>.
- [14] 程魁, 马良. 0-1 背包问题的萤火虫群优化算法[J]. 计算机应用研究, 2013, 30(4): 993-994.
- [14] LI Dan-dan, LEE D L. A topology-based semantic location model for indoor applications[C]//Proc of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM Press, 2008: 60-69.
- [15] LI Dan-dan, LEE D L. A lattice-based semantic location model for indoor navigation[C]//Proc of the 9th International Conference on Mobile Data Management. Washington DC: IEEE Computer Society, 2008: 17-24.
- [16] AFYOUNI I, RAY C, CLARAMUNT C. Spatial models for indoor and context-aware navigation systems; a survey[J]. Journal of Spatial Information Science, 2012, 4(1): 85-123.
- [17] BECKER T, NAGEL C, KOLBE T H. A multilayered space-event model for navigation in indoor spaces[M]//3D Geo-Information Sciences. Berlin: Springer, 2009: 61-77.
- [18] ZHU Qing, HU Ming-yuan. Semantics-based 3D dynamic hierarchical house property model[J]. International Journal of Geographical Information Science, 2010, 24(2): 165-188.
- [19] 高露雄, 王汇. 基于多目标决策与层次分析法的最佳路径模型[J]. 测绘与空间地理信息, 2011, 34(5): 221-224.
- [20] 王世明, 邢建平, 张玉婷, 等. 典型城市路网中的椭圆最短路径算法[J]. 系统工程理论与实践, 2011, 31(6): 1158-1164.
- [21] 陆锋. 最短路径算法: 分类体系与研究进展[J]. 测绘学报, 2001, 30(3): 269-275.
- [22] 方慧, 洪卫军. 城市应急救援中的动态限制搜索区域最短路径算法研究[J]. 中国人民公安大学学报: 自然科学版, 2010, 16(4): 87-91.

(上接第 984 页)

- [6] HARLE R. A survey of indoor inertial positioning systems for pedestrians[J]. Communications Surveys & Tutorials, 2013, 15(3): 1281-1293.
- [7] FISCHER C, GELLERSEN H. Location and navigation support for emergency responders; a survey[J]. IEEE Pervasive Computing, 2010, 9(1): 38-47.
- [8] KARIMI H A. Universal navigation[M]//Universal Navigation on Smartphones. New York: Springer, 2011: 75-88.
- [9] SWOBODZINSKI M, RAUHAL M. An indoor routing algorithm for the blind; development and comparison to a routing algorithm for the sighted[J]. International Journal of Geographical Information Science, 2009, 23(10): 1315-1343.
- [10] MUNZER S, STAHL C. Providing individual route instructions for indoor wayfinding in complex, multi-level buildings[C]//Proc of the 5th Geographic Information Days: Young Researchers Forum. 2007: 241-246.
- [11] GOETZ M, ZIPF A. Formal definition of a user-adaptive and length-optimal routing graph for complex indoor environments[J]. Geo-Spatial Information Science, 2011, 14(2): 119-128.
- [12] SCHOUGAARD K R, GRONBAK K, SCHARLING T. Indoor pedestrian navigation based on hybrid route planning and location modeling[C]//Proc of the 10th International Conference on Pervasive Computing. Berlin: Springer-Verlag, 2012: 289-306.
- [13] YUAN Wen-jie, SCHNEIDER M. iNav: an indoor navigation model supporting length-dependent optimal routing[M]//Geospatial Thinking. Berlin: Springer, 2010: 299-313.